<u>トップページ</u> -> 研究道具箱と教材 -> リレーショナルデータベース入門(実践で学ぶ) -> SQL 問い合わせ [サイトマップへ] [全文検索へ] [統計情報へ]

SQL 問い合わせ

URL: http://www.db.is.kyushu-u.ac.jp/rinkou/addb/3.html

データベース内の1つのテーブルに対する SQL 問い合わせの例. 問い合わせの結果として, 新しいテーブルが 1つ生成される.

| | id | product_name | type | cost |
|-------------------------------------|-------------------------------|---------------------|-----------|------|
| | | Fukuoka apple | apple | 50 |
| table 'products' (in a database) | | Kumamoto orange L | orange | 30 |
| | | Kumamoto orange M | orange | 20 |
| SQL query | 4 | Fukuoka melon | melon | NULL |
| SELECT * | | | | |
| | | | | |
| WHERE type = ' | ora | nge'; | | |
| | id | product_name | type | cost |
| | 2 | Kumamoto orange L | orange | 30 |
| | | Kumamoto orange M | orange | 20 |
| | | Query result is one | new table | |
| | | | 1 | |
| | id | product_name | type | cost |
| | 1 | Fukuoka apple | apple | 50 |
| table 'products' | 2 | Kumamoto orange L | orange | 30 |
| (in a database) | 3 | Kumamoto orange M | orange | 20 |
| SQL query | 4 | Fukuoka melon | melon | NULL |
| SELECT * FROM products | | | | |
| WHERE COST > 2 | ; C | | | |
| | id | product_name | type | cost |
| | 1 | Fukuoka apple | apple | 50 |
| | 2 | Kumamoto orange L | orange | 30 |
| | Query result is one new table | | | |

SQL 問い合わせ

| | id | product_name | type | cost |
|----------------------|------------------|-------------------|---------|------|
| | 1 | Fukuoka apple | apple | 50 |
| table 'products' | 2 | Kumamoto orange L | orange | 30 |
| (in a database) | 3 | Kumamoto orange M | orange | 20 |
| | 4 | Fukuoka melon | melon | NULL |
| SQL query | | · | | |
| | | product_name | cost | |
| | Fukuoka apple 50 | | | |
| Kumamoto orange L 30 | | | | |
| Kumamoto orange M 20 | | | | |
| | | Eukuaka malan | NILLI I | |
| | | Fukuoka meion | NULL | |



結合問い合わせの例



http://133.5.18.167/rinkou/addb/3.html



図. 条件 (condition) R.B = S.B

| SELECT * | |
|------------------|---|
| FROM R, S | |
| WHERE R.B = S.B; | |
| | - |

| А | R.B | S.B | С | D |
|---|-----|-----|---|---|
| а | b | b | С | f |
| а | d | d | е | а |
| а | d | d | е | С |

Join Query Example

Query Result is One Table

図.結合問い合わせの例



| Α | R.B | S.B | С | D |
|---|-----|-----|---|---|
| а | d | d | е | а |
| а | d | d | е | С |

Join Query Example

Query Result is One Table

図. 結合問い合わせの例(2)

If you want to get only the two attributes A and D

SELECT A, D FROM R, S WHERE R.B = S.B;

| Α | D |
|---|---|
| а | f |
| а | а |
| а | С |

Join Query Example

Query Result is One Table

図. 結合問い合わせの例(3)

SQL の比較演算子

比較演算子は2項述語である. <u>SQL</u>で,数値や文字列の比較演算子には次のようなものがある

∘ < ∘ <= ∘ =

• >=

• > • <>

SQL の論理演算子

AND, OR, NOT

· IS NULL

NULL 値を持つような行を得るために使うことができる.

• ¥'

「¥'」は、「'」を含むような文字列を扱うための貴方

・LIKEと%と_

SQLでの文字列のマッチングに使うキーワード

SQL 問い合わせ (SQL query)

<u>SQL</u> 問い合わせは, <u>リレーショナルデータベース</u>に格納された1つまたは複数の<u>テーブル</u>を使う. 問い合わせの評価 結果として, <u>リレーショナルデータベース</u>内の<u>テーブル</u>をそのままの形で得るということはめったになく, **新しい<u>テ</u>ーブ ルが 1つ作られる**.

<u>SQL</u> 問い合わせのプログラムは、SELECT、FROM、WHERE を使うのが基本である. 文法は次のようになる. ※ SQL の文法は多彩なので、ここでは、基本的な場合に説明を絞る.

SELECT <**list-of(result-column)**> FROM <li**st-of(table-name)**>

あるいは

SELECT <list-of(result-column)> FROM <list-of(table-name)> WHERE <expression>

例えば,

SELDCT A1, A2, ..., An FROM T1, T2, ..., Tm WHERE <expression>

のように書くと, m 個の<u>テーブル</u> T1, T2, ..., Tm の直積集合から <expression> を満足する行のみを選び, そうして出 来た<u>テーブル</u>の属性 A1, A2, ..., An を出力するという意味になる.

WHERE の部分が無い場合, 例えば,

 SELDCT
 A1,
 A2,
 ...,
 An

 FROM
 T1,
 T2,
 ...,
 Tm

のように書くと, m 個の<u>テーブル</u> T1, T2, ..., Tm の直積集合から属性 A1, A2, ..., An を出力するという意味になる.

SELECT <list-of(result-column)>

SELECT の後には、問い合わせの結果として得たい**列名あるいは列名を含む式の、1個以上の並び**を 書く、2個以上ある場合には半角のカンマ「、Jで区切る、問い合わせ結果として得られる<u>テーブル</u>の中の 列の並びは、<u>SQL</u>の SELECT に書いた列名あるいは列名を含む式の並びの順序通りになる.

SELECT の後に列名を書くとき、「<テーブル名>、列名>」のように、「<テーブル名>」を前に付けねばならない場合がある.これは、FROM に指定するテーブルが2個以上あり、しかも、これらのテーブルが同じ属性名の属性を持つ場合である。例えば、「科目」と「履修」という2つのテーブルが、同じ属性名「科目番号」を持つ場合、下記のように「科目番号」の前に「科目」」を付けるなどして、あいまいさを排除する必要がある.

SELECT 科目.科目番号,科目名 FROM 科目,履修 WHERE 科目.科目番号 = 履修.科目番号 AND 履修.学生番号 = 00001;

列名として「*」あるいは「<テーブル名>.*」のような書き方ができ、これは、全ての属性名を並べて書いたのと同じ意味を持つ。

• FROM <list-of(table-name)>

FROM の後には、テーブル名の1個以上の並びを書く、2個以上ある場合には半角のカンマ「,」で区切る.

テーブル参照リストの中で, タップル変数を定義することもできる. 例えば, 下記の <u>SQL</u> 文では X と Y の 2つのタップル変数が定義されている.

SELECT X. 社員番号, Y. 社員番号 FROM 社員 X. 社員 Y WHERE X. 部長 = Y. 社員番号 AND X. 給与 > Y. 給与

• WHERE <expression>

WHERE の後には**探索条件**を書く. 複数の探索条件を組み合わせたい場合には,(半角のカンマではなく)論理演算の AND, OR を用いる. 探索条件は, FROM で指定したテーブルの属性名や, FROM で指

定したタップル変数の変数名を含む式である. 探索条件に使用できるキーワードとしては次のものがある.

- 比較演算(<, <=, =, >=, >, <>)
- 論理演算(AND, OR, NOT)
- 各種の述語(BETWEEN, IN, LIKE, NULL, EXIST など)

など

θ 結合演算

R(A1, A2, …, An)とS(B1, B2, …, Bm)を<u>リレーション</u>とする. Rの<u>属性</u> AiとSの<u>属性</u> Bj 上の *θ* - 結合演算を「R[Ai θ Bj]S」と書く. 定義は次の通りである. (θ は比較演算子である).

R[Aiθ Bj]S は, RとSの直積集合 (R×S)の中から「RAiθ S.Bj」を満足する要素を選んだもの

 θ -結合演算 [R[Ai θ Bj]S」を SQL を用いて書くと次の通りである.

SELECT * FROM R, S WHERE <Ai**Ø**Bj>

※ (参考)リレーショナル代数式を使って次のように書くことができる.

 $R[Ai \theta Bj]S = \{ (t, u) \mid t \in R \land u \in S \land t[Ai] \theta u[Bj] \}$

但し、「(t, u)」と書いているのは、t = (a1, a2, …, an), u = (b1, b2, …, bm) とするときに、(t, u) = (a1, a2, …, an, b1, b2, …, bm) なる n + m 項のタップルである.

・結合問い合わせ(結合質問ともいう)

結合問い合わせとは、問い合わせの中に θ-結合演算を含むような問い合わせのことである.結合問い合わせでは、 SQL のテーブル参照リストに、2つ以上のテーブル名が現れる.一方で、結合問い合わせの評価結果は、一般の問い 合わせと同様に1つの<u>テーブル</u>であることに注意して欲しい.

・テキスト・エンコーディング (text encoding)

データベースには、文字データはエンコード (encoding) されて格納されている. エンコードの手法にはいくつかの種類がある. SQLite バージョン 3 では、エンコードとして次のいずれかを指定できる.

- UTF-8
- ∘ big-endian UTF-16
- ∘ little-endian UTF-16

演習

演習で行うこと

- ・ <u>SQLiteman の起動と終了</u> (Start and end SQLiteman)
- <u>SQLiteman で新しいデータベースを作成する</u>(Create a new database)
- · <u>SQLiteman で既存のデータベースを開く</u>(Open an existing database using SQLiteman)
- <u>SQL を用いたテーブル定義と一貫性制約の記述</u> (Table definition and integrity constraint specification using SQL)

リレーショナル・スキーマ (relational schema): **products(id, product_name, type, cost, created_at)**

```
SQL プログラム:
```

| CREATE TABLE prod | ucts (| |
|--------------------------|----------|-------------------------------------|
| id | INTEGER | PRIMARY KEY AUTOINCREMENT NOT NULL, |
| product_name | TEXT | UNIQUE NOT NULL, |
| type | TEXT | NOT NULL, |
| cost | REAL, | |
| created_at | DATETIME | NOT NULL); |

<u>SQL を用いたテーブルへの行の挿入</u> (Insert rows into a table using SQL)

```
BEGIN TRANSACTION:
```

```
INSERT INTO products VALUES(1, 'Fukuoka apple', 'apple', 50, datetime('now'));
INSERT INTO products VALUES(2, 'Kumamoto orange L', 'orange', 30, datetime('now'));
INSERT INTO products VALUES(3, 'Kumamoto orange M', 'orange', 20, datetime('now'));
INSERT INTO products VALUES(4, 'Fukuoka melon', 'melon', NULL, datetime('now'));
COMMIT;
```

・ <u>SQL 問い合わせの発行と評価結果の確認</u> (Issue SQL queries and inspect the results)

```
SELECT * FROM products:
SELECT * FROM products WHERE type = 'orange';
SELECT * FROM products WHERE cost > 25;
```

<u>SQL を用いたテーブル定義と一貫性制約の記述</u> (Table definition and integrity constraint specification using SQL)

リレーショナル・スキーマ (relational schema):

requests(id, year, month, day, product_id, qty, created_at)

```
SQL プログラム:
```

```
      CREATE TABLE requests (

      id
      INTEGER
      PRIMARY KEY AUTOINCREMENT NOT NULL,

      year
      INTEGER
      NOT NULL CHECK ( year > 2008 ),

      month
      INTEGER
      NOT NULL CHECK ( month >= 1 AND month <= 12 ),</td>

      day
      INTEGER
      NOT NULL CHECK ( day >= 1 AND day <= 31 ),</td>

      product_id
      INTEGER
      NOT NULL REFERENCES products(id),

      qty
      INTEGER
      NOT NULL,

      created_at
      DATETIME
      NOT NULL );
```

- ・<u>SQL を用いたテーブルへの行の挿入</u> (Insert rows into a table using SQL)
- ・ <u>SQLiteman を用いたデータのブラウズ</u> (Browse Data using SQLiteman)
- <u>SQL 問い合わせの発行と評価結果の確認</u> (Issue SQL queries and inspect the results)
- <u>SQL を用いたテーブル定義と一貫性制約の記述</u> (Table definition and integrity constraint specification using SQL)
 リレーショナル・スキーマ (relational schema, created_at):

bundles(id, request_id, qty, shipping_id)

shippings(id, year, month, day, created_at)

SQL プログラム:

- ・ <u>SQL を用いたテーブルへの行の挿入</u> (Insert rows into a table using SQL)
- ・ <u>SQLiteman を用いたデータのブラウズ</u> (Browse Data using SQLiteman)
- <u>SQL 問い合わせの発行と評価結果の確認</u> (Issue SQL queries and inspect the results)

演習を行うために必要になる機能や文法

SQLite の SQL の説明は <u>http://www.hwaci.com/sw/sqlite/lang.html</u> (English Web Page)にある.

```
- SQL 問い合わせ (SQL query)
```

SELECT <list-of(result-column)> FROM <list-of(table-name)>

table-name で指定したテーブルの直積集合から, list-of(result-column) で指定した属性を抽出したり, 式を評価して新しいテーブルを1つ作り出力する.

SELECT <list-of(result-column)> FROM <list-of(table-name)> WHERE <expression>

```
table-name で指定したテーブルの直積集合から、 (expression)を満足する行のみを選び, そうして出
来たテーブルから, list-of(result-column)で指定した属性を抽出したり, 式を評価して新しいテーブルを
1つ作り出力する.
```

SQLiteman の起動と終了 (Start and end SQLiteman)

CREATE TABLE などの SQL コマンドを編集できるエディタの機能等をもったソフトウエアとshちえ SQLiteman を使うことにします. (We use the SQLiteman as SQL editor, database manager interface, ...)

1. SQLiteman の起動 (Start SQLiteman)

■ Ubuntu での SQLiteman の起動例

「プログラミング」→「Sqliteman」と操作する.



SQLiteman の新しいウインドウが開く(A New window appears)

| Sche <u>m</u> a <u>P</u> ragmas | | |
|---------------------------------|-----------------------|----------------|
| Database | 1 | |
| | | |
| | | |
| | | C 1 4 D 4 |
| | | Col: 1 Row: 1, |
| | | |
| | Full View Item View S | cript Output |
| | | |
| | 1 | |
| | | |
| | | |

「SQLiteman」のアイコンをダブルクリック (double click "SQLiteman.exe")



SQLiteman の新しいウインドウが開く(A New window appears)

| a Duna Duna Duna De | |
|--|-----------------|
| Scheme Programs | |
| Dotabase 1 | |
| a de la constante de la consta | Col: 1 Boy: 1/1 |
| | |
| Full View Item View Script Output | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |

2. ヘルプの表示 (Help Content)

 $\lceil \mathsf{Help} \rfloor \rightarrow \lceil \mathsf{Help} \ \mathsf{Content} \rfloor$

| che <u>m</u> a <u>P</u> ragmas | Help Content F1 | |
|---|---|----------------------------------|
| Database | | |
| Soliteman Help | | |
| | | |
| | The Sqliteman Handbook | Nevt |
| <pre>< petr@scribus.info> 1.1</pre> | | |
| Copyright © 2007 Pe This documentation is r | tr Vanek released under GNU Free Documentation License. S | see <u>Credits</u> for more. |
| Sqliteman is the best joking here (or just a | developer's and/or admin's GUI tool for Sqlite3 in bit only) – it contains the most complete feature se | the world. No et of all tools |
| available. | | |
| Table of Contents | | |

3. **終了** (End SQLiteman) 「**File**」→「**Exit**」で終了.

| <u>N</u> ew | Ctrl+N | | Q |
|--------------------------|--------|---|---|
| <u>0</u> pen | Ctrl+O | | |
| <u>R</u> ecent Databases | | 1 | |
| Preferences | | | |
| E <u>x</u> it | Ctrl+Q | | |
| | | | F |

SQLiteman で新しいデータベースを作成する (Create a new database)

以下の手順で, **新しいデータベースを作成する**. その結果, データベースファイルができる. (Create a new database)

1. 「File」→「New」

| Sqliteman | |
|-----------------------------------|---|
| File Context Database System Help | |
| New Ctrl+N | |
| Open Ctrl+O | |
| Recent Databases | 1 |
| Preferences | |
| 🖾 E <u>x</u> it Ctrl+Q | |
| | |
| | |
| | |

2. データベースの新規作成を開始する (Start a new databae creation)

■ Ubuntu での実行例(データベースファイル名を「SQLite/mydb」にする場合)

データベースファイル名 SQLite/mydb を指定し、「保存」をクリック (Click '保存' after writing a database file name)

データベースを新規作成したいときは、データベースファイル名として「新しい」ものを指定すること

| 💌 🗔 New Database | | |
|------------------------|-------------|---------------------------|
| 名前(<u>N</u>): | SQLite/mydb | |
| フォルダの中に保存(<u>F</u>): | windowslike | ▼ |
| ▷ 他のフォルダ(<u>B</u>) | | |
| | | ※キャンセル(● 保存(S) ● |

3. **データベースの中身が表示されるので確認する** (Database appears) このときテーブル (Tables) 数も, ビュー (Views) の数も 0 である. 4.

| <u>File Context D</u> atabase <u>S</u> ystem <u>H</u> elp | |
|---|-----------------------------------|
| Sche <u>m</u> a <u>P</u> ragmas | : 🕨 🧣 🍽 🛅 🗁 🏝 🖹 🔍 |
| Database ▼ ■ main III Tables (0) III Tables (0) <td></td> | |
| | Col: 1 Row: 1/1 |
| | |
| | Full View Item View Script Output |
| | Query OK Row(s) returned: 0 |
| テキスト・エンコーディングの設定を確認 する (text enco まず,「 Pragmas 」をクリック. (Click 'Pragmas') | ding) |
| File <u>Context</u> | <u>System</u> <u>H</u> elp |

| File | Context Database System Help |
|------|------------------------------|
| Sch | nema Pragmas |
| D | atabase |
| ~ | 🕫 😂 main |
| | III Tables (0) |
| | 📼 Views (0) |
| | 🕨 🥡 System Catalogue (1) |
| | |

encodingの行に「UTF-8」のように表示されている.

| | Name | Value | |
|----------|-----------------------|---------|--|
| 1 | auto vacuum | 0 | |
| 2 | cache size | 2000 | |
| 3 | case sensitive | Not Set | |
| 4 | count_changes | 0 | |
| 5 | default_cache | 2000 | |
| 6 | default_synch | Not Set | |
| 7 | empty result | 0 | |
| 8 | encoding | UTF-8 | |
| 9 | full_column_n | 0 | |
| 10 | fullfsync | 0 | |
| 11 | lenacy file fo | 1 | |
| Det P | ail Pragma: | | |
| | auto_vacuum | | |
| | /alue: | | |

※ もし, **データベースの文字のエンコーディングを変えたい**ときは、 SQLiteman のようなグラフィカルなツールを使う のではなく、<u>salite.exe</u>を起動し「**PRAGMA encoding=…**;」で変える方がずっと簡単でしょう. 例えば「UTF-16le」などに 変えたいなど.

5. 終了 (End SQLiteman)

あとの混乱を防ぎたいので、1度、SQLiteman を終了する

「File」→「Exit」で終了.

| <u>N</u> ew | Ctrl+N | |
|-------------------------|--------|--|
| 🥌 <u>O</u> pen | Ctrl+O | |
| <u>R</u> ecent Database | s 🔰 📕 | |
| Preferences | | |
| 🖾 Exit | Ctrl+Q | |

SQLiteman で既存のデータベースを開く(Open an existing database using SQLiteman)

すでに作成済みのデータベースを、下記の手順で開くことができる.

以下の手順で, 既存のデータベースファイルを開く (Open an existing database file)

1. $[File] \rightarrow [Open]$

| File Context Database System Help | |
|--|-----------------|
| New Ctrl+N Open Ctrl+O Recent Databases Preferences | |
| E <u>x</u> it Ctrl+Q | Col: 1 Row: 1/1 |

- 2. データベースファイルを開く(Open Database File)
 - Ubuntu での実行例(「SQLite/mydb」を開く場合)

データベースファイル **SQLite/mydb** を選び,「開く」をクリック (Click '開く' after choosing the database file "SQLite/mydb")

| 🔛 🗔 Open Database | | | | | X |
|----------------------------------|------------------|----------------|------------------|----------------|---|
| 🖉 🖵 home 🗟 wind | dowslike SQLite | | | | |
| 場所(P) | 名前 | - | サイズ | 最終変更日 | |
| 🔍 検索 | 📄 mydb | | 0パイト | 17:18 | |
| 🕗 最近開いたファイル | sqlite3_analyzer | | 1.7 MB | 2010年08月24日 | |
| 🗟 windowslike | | | | | |
| 🗐 デスクトップ | | | | | |
| | | | | | |
| フロッピー・ドライブ | | | | | |
| ドキュメント | | | | | |
| ■ 西海 | | | | | = |
| □回家 □ □ □ □ | | | | | |
| ■ ダウンロード | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | _ | | - |
| ◆追加(<u>A</u>) ──削除(<u>R</u>) | | | S | QLite database | - |
| | | O + | | | |
| | | 2 + + 2 | 2011(<u>C</u>) | (□) | |

■ Windows での実行例(「C:¥SQLite¥mydb」を開く場合)

データベースファイル C:**¥SQLite¥mydb** を選び,「開く」をクリック(Click '開く' after choosing the database file "C:¥SQLite¥mydb")

要するに、/home/<ユーザ名>/SQLite の下の mydb を選ぶ.

| Open Database | | | | <u>? ×</u> |
|---------------------|------------------------|-----|-------|------------|
| ファイルの場所型 () SQLite | \supset | • + | 🗈 💣 🎫 | |
| alite3 | | | | |
| 最近使ったファイル | | | | |
| | | | | |
| デスクトップ | | | | |
| | | | | |
| | | | | |
| マイ コンピュータ | | | | |
| | | | | |
| マイ ネットワーク | | | | |
| ファイル名(M |): | | · (| 關⟨◎⟩ |
| ファイルの種 | 類①: SQLite database (約 | k) | • | キャンセル |

- 3. データベースの中身が表示されるので確認する (Database appears)
 - ◆ 表示例(データベースの中身によって表示が変わる)

| S mydb – Sqliteman | |
|--|---|
| <u>File Context</u> <u>Database</u> <u>System</u> <u>H</u> elp | |
| Sche <u>m</u> a <u>P</u> ragmas | |
| | |
| Database | |
| 🖻 📮 main | |
| ⊕- 111 Tables (2) | |
| views W/ | |
| 🗄 🐨 🕎 System Catalogue (2) | |
| |) |
| | |
| | |
| | |

4 「Tables」の数字が1以上の場合には展開できる

「Tables」の数字が1以上のとき、「Tables」を展開すると、テーブルの一覧 (List of Tables) が表示される

ので確認する (List of tables appears by clicking 'Tables')

◆ 展開の例



5. **テキスト・エンコーディングの設定を確認**する (text encoding) まず、「**Pragmas**」をクリック. (Click 'Pragmas')



encodingの行に「UTF-8」のように表示されている.

| ſ | Sche | e <u>m</u> a <u>P</u> ragmas | | |
|---|------|------------------------------|------------|---|
| | | Name | Value | |
| | 1 | auto_vacuum | 0 | |
| | 2 | cache_size | 2000 | |
| | 3 | case_sensitive | Not Set | |
| | 4 | count_changes | 0 | |
| | 5 | default_cache_size | 2000 | |
| | 6 | default_synchro*** | Not Set | |
| | 7 | empty_result_ca··· | 0 | |
| (| 8 | encoding | UTF-8 |) |
| | 9 | full_column_names | 0 | |
| | 10 | fullfsync | 0 | |
| | 11 | legacy_file_format | 1 | |
| | 12 | locking_mode | normal | |
| | 13 | page_size | 1024 | |
| | 14 | max_page_count | 1073741823 | |
| | 15 | read_uncommitted | 0 | |
| | 16 | short_column_n… | 1 | |
| | 17 | synchronous | 2 | |
| | 18 | temp_store | 0 | |
| | 19 | temp_store_dire… | Not Set | |

※ もし, **データベースの文字のエンコーディングを変えたい**ときは、 SQLiteman のようなグラフィカルなツールを使う のではなく、<u>sqlite.exe</u>を起動し「PRAGMA encoding=...;」で変える方がずっと簡単でしょう. 例えば「UTF-16le」などに 変えたいなど.

6. 「System Catalogue」を展開し、「sqlite_master」をクリックすると、データベース・スキーマ (database schema) が表示される ので確認する (Database schema appears by clicking 'sqlite_master')

◆ 表示の例(1)

データベースが空の場合,表示も空.

| Database | 1 |
|---------------|-----------------------------------|
| | |
| sqlite_master | Col: 1 Row: |
| | |
| | Full View Item View Script Output |
| | type name tbl_name rootpage so |
| | Query OK |

score_records, order_records などのテーブルを定義すみの場合

| | | | | | |
|--|------|--------|----------------------------------|---------------|--|
| Sche <u>m</u> s Progmos | Þ | 2 | III 🖻 🖻 🚰 🔔 🤅 | <u>}</u> | |
| Database Image: mein Image: Tables (2) Image: Tables (2) </th <th>1</th> <th></th> <th></th> <th></th> <th></th> | 1 | | | | |
| | Ful | l View | Itom View Script Dutput | | |
| | Г | type | name | tbl_name | rootpage |
| | 1 | Itable | scoro_rccords | score_records | 2 CREATE TABLE score records (rame TEXT NOT NUL) score INTEGER NOT NULL stucent_name TEXT NOT NULL createc_at DATETIME NOT NUL updated_at DATETIME, UNIQUE (name, student_name)) |
| | 2 | index | solite_autoindex_score_records_1 | score_records | 3 fnulli |
| | | tablo | ordor_ropords | orda_records | 4 CREATE TABLE ordor procords (id INTEGER PRIMARY year INTEGER NOT NUL month INTEGER NOT NUL day INTEGER NOT NUL customer power ITAT NOT NUL |
| | 3 | | | | productyname TEXT NOT NULL unit price REAL NOT NULL qty INTEGER NOT NULL created_at DATEIIME NOT N upJateJ_at CATETIME, CHECK (unit price * cty) < 2 |

SQL を用いたテーブル定義と一貫性制約の記述 (Table definition and integrity constraint specification using SQL) SQL を用いて, products テーブルを定義し、一貫性制約を記述する. (Define 'products' table and specify integrity constrants of the table using SQL) リレーショナル・スキーマ (relational schema): products (id, product_name, type, cost, created_at)

1. products テーブルの定義 (Define a table)

次の SQL を入力し,「Run SQL」のアイコンをクリック (Write the following SQL, and click "Run SQL" icon).

| CREATE TABLE p rod i | l cts (| | |
|-----------------------------|----------------|-------------------------------|------|
| id | INTEGER | PRIMARY KEY AUTOINCREMENT NOT | NULL |
| product_name | TEXT | UNIQUE NOT NULL, | |
| type | TEXT | NOT NULL, | |
| cost | REAL, | | |
| created_at | DATETIME | NOT NULL): | |

※「SQL Editor」のウインドウには、SQL プログラムを書くことができる. In the 'SQL Editor' window, you can write down SQL program(s).

| 🥌 mydb - Sqliteman | |
|--|--|
| <u>File Context Database System Help</u> | |
| Sche <u>m</u> a <u>P</u> ragmas | 🜔 😰 🖿 📄 🖬 🗠 🔕 |
| Database | 1 SCREATE TABLE products (ic 2 ic 3 product,name TEXT 4 type 5 cost 6 UNIQUE NOT NULL, orea.ed_at |

2. コンソールの確認 (Inspect console)

```
エラーメッセージが出ていないことを確認
```

SQL を用いたテーブルへの行の挿入 (Insert rows into a table using SQL)

次のような products テーブルを作る. (Construct table 'products')

| id | product_name | type | cost |
|----|-------------------|--------|------|
| 1 | Fukuoka apple | apple | 50 |
| 2 | Kumamoto orange L | orange | 30 |
| 3 | Kumamoto orange M | orange | 20 |
| 4 | Fukuoka melon | melon | NULL |

以下の手順で, SQL を用いて products テーブルへの行の挿入を行う (Insert rows into table 'products' using SQL)

1. SQL プログラムの記述

「INSERT INTO ...」は行の挿入. ここには 4つの SQL 文を書き、「BEGIN TRANSACTION」と「COMMIT」で囲む. ("INSERT INTO ..." means inserting a row into a table. Four SQL statements are wrote).

※ つまり, 挿入の前に BEGIN TRANSACTION; を実行し, 一連の挿入が終わったら COMMIT; を実行する. (Issue "BEGIN TRANSACTION" before database update and "COMMIT" after database update).

| BEGIN | TRANSACTION ; | |
|--------|---------------|--|
| THOFPT | 11170 | |

| DECEM TRANSACTION. INSERT INTO products VALUES(1, 'Fukuoka apple', 'apple', 50, datetime('now')); INSERT INTO products VALUES(2, 'Kumamoto orange L', 'orange', 30, datetime('now')); INSERT INTO products VALUES(3, 'Kumamoto orange M', 'orange', 20, datetime('now')); INSERT INTO products VALUES(4, 'Fukuoka melon', 'melon', NULL, datetime('now')); COMMIT: | |
|---|-------|
| File Context Database System Help | |
| Schema Pragmas Database Image: Construct the second se | |
| INSERT INTO には 2つの方法がある. (Two styles of "INSERT INTO") | |
| ■ 属性の値を, テーブル定義の順に全て並べる (List all attribute values. The order is the same as its definition) | table |

INSERT INTO products VALUES(1, 'Fukuoka apple', 'apple', 50, datetime('now')); ■ 属性の値の並び方を, 属性名を使って明示的に指定する (Specify the order of attribute values using attribute name list)

このとき、属性値を省略すると、テーブル定義のときに指定されたデフォルト値が使われる (defaults values are used)

INSERT INTO products VALUES(2, 'Kumamoto orange L', 'orange', 30, datetime('now'));

2. 複数の SQL 文の一括実行 (Execute multiple SQL statements)

複数の SQL 文を一括実行したいので、**カーソルを先頭行に移動**した後に、「Run multiple SQL statements ...」のボタ ンをクリックする.「Move the cursor to the top statement. Click ["]Run multiple SQL statements from current cursor position in one batch" icon)

| S | mydb – Sqliteman | | |
|----|--|-----|--|
| Ei | le <u>C</u> ontext <u>D</u> atabase <u>S</u> yst | tem | Help |
| | Sche <u>m</u> a <u>P</u> ragmas | | |
| | Database | | 1 BEGIN TRANSACTION |
| | 🖻 🖳 main | | 2 INSERT INTO products VALUES(1, 'Fukuokia apple', 'apple', bU, datetime('now')); |
| | | | 3 INSERT INTO products VALUES(2, 'Kumamoto orange L', 'orange', 30, datetime('now')); |
| | | | 4 INSERT INTO products VALUES(3, 'Kumamoto orange M', 'orange', 20, datetime('now')); |
| | 📄 🦳 🧰 Views (0) | | 5 INSERT INTO products VALUES(4, 'Fukuoka melon', 'melon', NULL, datetime(now')); |
| | 🗄 🌍 System Ca… | | 6 COMMIT; |

3. 「Script Output」ウインドウの確認 (Inspect "Script Output" window)

エラーメッセージが出ていないことを確認

| Full Vi | iew Item View Script Output | |
|----------|--|--|
| | Script started | |
| 2 | E BEGIN TRANSACTION; | |
| 3 | | |
| 5 | INSERT INTO products VALUES(1, 'Fukuoka apple', 'apple', 50, datetime(now')); | |
| | | |
| 8 | INSERT INTO products VALUES(2, 'Kumamoto orange L', 'orange', 30, datetime('now')); No error | |
| 10 | | |
| 11 12 | INSERT INTO products VALUES(3, 'Kumamoto orange M', 'orange', 20, datetime('now')); No error | |
| 13 | | |
| 14 | INSERT INTO products VALUES(4, 'Fukuoka melon', 'melon', NULL, datetime('now')); | |
| 15 | No error | |
| 16 | | |
| 17 | COMMIT; | |
| 18 | No error | |
| 19 | | |
| 20 | Script finished | |
| 121 | | |

Ж

エラーメッセージが出ているときは、テーブル定義を書き直して、「Run multiple SQL statements …」のボタンをクリック する、

例えば、下記のようにエラーが出ていたとする. このときは、

- 「BEGIN TRANSACTION」は終わっている。
- 。それ以降は実行されていない



このようなときは、再開したい行にカーソルをあわせて、「Run multiple SQL statements ...」のボタンをクリックする.

SQL 問い合わせの発行と評価結果の確認 (Issue SQL queries and inspect the results)

ここでは, SQL を用いた**問い合わせ**の実行例を示す. SQL 問い合わせの詳細については, <u>別の Web ページ</u>で説明する. ここでは, テーブルの中身を確認して欲しい.

テーブルの全ての行の表示 (List all rows of a table)

SELECT * FROM products:

| I 🕑 🕅 🕪 | 📾 📑 🗁 🏝 🏝 | | | |
|-------------------|-------------------------|----------|--------|---------------------|
| 1 SELEC | FROM products; | | | Þ |
| Duration: 0 secon | ls | | | * Col: 24 Row: 1/1 |
| | 🕈 🎽 🛛 🏹 🗍 | . | | |
| Full View I | em View Script Output | | | |
| id | product_name | type | cost | created_at |
| 1 | 1 Fukuoka apple | apple | 50 | 2009-12-10 07:16:00 |
| 2 | 2 Kumamoto orange L | orange | 30 | 2009-12-10 07:16:00 |
| 3 | 3 Kumamoto orange M | orange | 20 | 2009-12-10 07:16:00 |
| 4 | 4 Fukuoka melon | melon | {null} | 2009-12-10 07:16:00 |

条件を満足する行のみの表示 (List the rows which satisfy a given condition)

SELECT * FROM products WHERE type = 'orange':

| 🕨 🍘 🖬 🖕 🏝 | Q | | |
|-------------------------------------|---------------|------|---------------------|
| 1 SELECT * FROM products WHERE type | e = 'orange'; | | |
| | | | |
| | | | |
| | | | Ľ |
| Duration: 0 seconds | | | * Col: 46 Row: 1/1 |
| | 1 | | |
| Full View Item View Script Output | | | |
| id product_name | type | cost | created_at |
| 1 2 Kumamoto orange L | orange | 30 | 2009-12-10 07:16:00 |
| 2 3 Kumamoto orange M | orange | 20 | 2009-12-10 07:16:00 |

SELECT * FROM products WHERE cost > 25:

| | 9 | | | |
|-------------------------------------|---------|------|---------------------|-------|
| 1 SELECT * FROM products WHERE cost | t > 25; | | | |
| | | | | |
| Duration: 0.016 seconds | | | * Col: 40 Row | : 1/1 |
| 8 8 8 2 2 1 | - | | | |
| Full View Item View Script Output | | | | |
| id product_name | type | cost | created_at | |
| 1 1 Fukuoka apple | apple | 50 | 2009-12-10 07:16:00 | |
| 2 2 Kumamoto orange L | orange | 30 | 2009-12-10 07:16:00 | |

SELECT * FROM products WHERE cost > 18:

| > ? I 🕨 | 💼 📑 🗁 🙆 | 2 Q | | |
|---------------------|------------------------|--------------|------|---------------------|
| SELECT | * FROM products WHERE | cost > 18; | | |
| | | | | |
| | | | | |
| | | | | |
| ration: O second: | 3 | | | * Col: 40 Row: 1/1 |
| | 💣 🗳] 🂽 🕑 |] 🚭 | | |
| Full View 1te | m View Script Output | | | |
| id | product_name | type | cost | created_at |
| 1 1 | Fukuoka apple | apple | 50 | 2009-12-10 07:16:00 |
| 2 2 | Kumamoto orange L | orange | 30 | 2009-12-10 07:16:00 |
| 3 | Kumamoto orange M | orange | 20 | 2009-12-10 07:16:00 |

SQL を用いたテーブル定義と一貫性制約の記述 (Table definition and integrity constraint specification using SQL)



<u>実体関連図 (Entity Relationship Diagram)</u>

SQL を用いて, **requests テーブルを定義し, 一貫性制約を記述**する. (Define a table 'requests'. Specify integrity constrants of the table using SQL)

1. products テーブルの定義 (Define a table)

.....

次の SQL を入力し,「Run SQL」のアイコンをクリック (Write the following SQL, and click "Run SQL" icon).

| GREATE | TABLE | requests (| |
|--------|-------|------------|-----|
| i d | | INTEGED | DDI |

| id | INTEGER | PRII | MARY KEY AUTOINCREMENT NOT NULL, |
|------------|----------|------|--|
| year | INTEGER | NOT | NULL CHECK (year > 2008), |
| month | INTEGER | NOT | NULL CHECK (month >= 1 AND month <= 12), |
| day | INTEGER | NOT | NULL CHECK (day >= 1 AND day <= 31), |
| product_id | INTEGER | NOT | NULL REFERENCES products(id), |
| qty | INTEGER | NOT | NULL, |
| created at | DATETIME | NOT | NULL); |



2. コンソールの確認 (Inspect console)

エラーメッセージが出ていないことを確認



SQL を用いたテーブルへの行の挿入 (Insert rows into a table using SQL)

次のような requests テーブルを作る. (Construct table 'requests')

| request | 5 | | | | | |
|---------|------|-------|-----|------------|-----|--|
| d | year | month | day | product id | qty | |
| 1 001 | 2009 | 10 | 28 | 1 | 3 | |
| 1 002 | 2009 | 11 | 1 | 2 | 1 | |
| 1 003 | 2009 | 11 | 2 | 1 | 2 | |
| 1004 | 2009 | 11 | 2 | 3 | 4 | |

以下の手順で, SQL を用いて requests テーブルへの行の挿入を行う (Insert rows into table 'requests' using SQL)

1. SQL プログラムの記述

「INSERT INTO ...」は行の挿入. ここには **4つの SQL 文**を書き,「BEGIN TRANSACTION」と「COMMIT」で囲む. (″INSERT INTO ...″ means inserting a row into a table. Four SQL statements are wrote).

BEGIN TRANSACTION: INSERT INTO **requests** VALUES(1001, 2009, 10, 28, 1, 3, datetime('now')): INSERT INTO **requests** VALUES(1002, 2009, 11, 1, 2, 1, datetime('now')):

| INSERT INSERT COMMIT | INTO INTO | requests requests | VALUES(VALUES(| 1003, 1004, | 2009, 2009, | 11, 11, | 2, 2, | 1, 3, | 2, 4, | datetime('now')); datetime('now')); |
|----------------------------|--------------|----------------------|--------------------|----------------|----------------|------------|----------|----------|----------|--|
| | | | | | | | | | | |

| 😂 mydb – Sqliteman | |
|---|---|
| <u>File Context Database System H</u> elp | |
| Sche <u>m</u> a <u>P</u> ragmas | - 🛛 🕨 📼 🗀 🗁 🏝 🔍 |
| Database | Image: Description of the second se |

2. 複数の SQL 文の一括実行 (Run multiple SQL statements)

複数の SQL 文を一括実行したいので、カーソルを先頭行に移動した後に、「Run multiple SQL statements ...]のボタ ンをクリックする. 「Move the cursor to the top statement. Click "Run multiple SQL statements from current cursor position in one batch" icon)

| Smydb - Sqliteman | | | | | | | | | | | |
|--|--|--|--|--|--|--|--|--|--|--|--|
| <u>File Context Database System Hel</u> |) | | | | | | | | | | |
| Sche <u>m</u> a <u>P</u> ragmas | | | | | | | | | | | |
| Database | 1 BEGIN TRANSACTION | | | | | | | | | | |
| ⊡- 曼 main ⊕- 🏢 Tables (4) 📾 Views (0) ⊕- 😨 System Catalogue (2) | 2 INSERT INTO requests VALUES(1001, 2009, 10, 28, 13, datetime(now)); 3 INSERT INTO requests VALUES(1002, 2009, 11, 1, 2, 1, datetime(now)); 4 INSERT INTO requests VALUES(1003, 2009, 11, 1, 2, 1, datetime(now)); 5 INSERT INTO requests VALUES(1003, 2009, 11, 2, 1, 2, datetime(now)); 6 COMMIT; COMMIT; | | | | | | | | | | |

- 3. 「Script Output」ウインドウの確認 (Inspect "Script Output" window)
 - エラーメッセージが出ていないことを確認

| Full View Item View Script Output | |
|--|--|
| 1 Script started | |
| 2 BEGIN TRANSACTION; 3 No error | |
| 4 5 INSERT INTO requests VALUES(1001, 2009, 10, 28, 1, 3, datetime(now)); | |
| 6 No error 7 | |
| 8 INSERT INTO requests VALUES(1002, 2009, 11, 1, 2, 1, datetime(now)); 9 No error | |
| 10 | |
| 12 No error | |
| 14 INSERT INTO requests VALUES(1004, 2009, 11, 2, 3, 4, datetime(now)); | |
| 15 No error 16 | |
| 17 COMMIT; 18 No error | |
| 19 20 Script finished | |
| ZI Competimistica | |

SQLiteman を用いたデータのブラウズ (Browse Data using SQLiteman)

· products テーブル

まず、オブジェクト・ブラウザ (Object Browser)の中の「Tables」を展開 (Click 'Tables')



次に、テーブル productsを選ぶ (Select table 'products')



テーブル **products**が表示される (table 'products' appears)

| F | Ful | l View 📄 Ite | em View Script Output | | | |
|---|-----|--------------|-----------------------|--------|--------|---------------------|
| ſ | _ | id | product_name | type | cost | created_at |
| | 1 | 1 | Fukuoka apple | apple | 50 | 2009-12-10 07:16:00 |
| ľ | 2 | 2 | Kumamoto orange L | orange | 30 | 2009-12-10 07:16:00 |
| ľ | 3 | 3 | Kumamoto orange M | orange | 20 | 2009-12-10 07:16:00 |
| | 4 | 4 | Fukuoka melon | melon | {null} | 2009-12-10 07:16:00 |

※ もし、データに間違いがあれば、このウインドウで修正できる (If you find any mistakes, you can modify the data using this window).

・今度は, **requests テーブル** を表示

| F | ul | l View 🛛 Iter | n View Sci | ript Output 📔 | | | | |
|---|----|---------------|--------------|---------------|-----|------------|-----|---------------------|
| ſ | | id | year | month | day | product_id | qty | created_at |
| | 1 | 1001 | 2009 | 10 | 28 | 1 | 3 | 2009-12-10 08:47:21 |
| | 2 | 1002 | 2009 | 11 | 1 | 2 | 1 | 2009-12-10 08:47:21 |
| | 3 | 1003 | 2009 | 11 | 2 | 1 | 2 | 2009-12-10 08:47:21 |
| U | 4 | 1004 | 2009 | 11 | 2 | 3 | 4 | 2009-12-10 08:47:21 |

SQL 問い合わせの発行と評価結果の確認 (Issue SQL queries and inspect the results)

直積 (Cartesian product)

| l | requests | | | | | | | product | 5 | | |
|---|----------|------|-------|-----|------------|-----|-----|---------|-------------------|--------|------|
| L | id | year | month | day | product id | qty |] [| id | product_name | type | cost |
| L | 1 001 | 2009 | 10 | 28 | 1 | 3 | | 1 | Fukuoka apple | apple | 50 |
| L | 1002 | 2009 | 11 | 1 | 2 | 1 | | 2 | Kumamoto orange L | orange | 30 |
| | 1003 | 2009 | 11 | 2 | 1 | 2 | | 3 | Kumamoto orange M | orange | 20 |
| L | 1004 | 2009 | 11 | 2 | 3 | 4 | | 4 | Fukuoka melon | melon | NULL |

SQL を使い、複数のテーブルの直積を簡単に得ることができる.

SELECT * FROM requests, products;

| 12 | SELECT * FROM requests, products; | | | | | | | | | | | | | |
|-----|---|------|------|-------------|----|------------|-----------------|---------------------|---|-------------------|--------|--------|---------------------|-----|
| | | | | | | | | | | | | | | |
| Dur | vuration: 0.016 seconds * Col: 25 Row: 2. | | | | | | | | | | | | | |
| | 33 | | | ľ | | | - | | | | | | | |
| F | Full View Item View Script Output | | | | | | | | | | | | | |
| ſ | - | | · | | | | | | | | | | | ᠷ |
| | 1 (| 1001 | 2009 | month 10 | 28 | product_id | <u>ціў</u> 3 | 2009-12-10 08:47:21 | 1 | Fukuoka apple | apple | 50 | 2009-12-10 07:16:00 | |
| | 2 | 1001 | 2009 | 10 | 28 | 1 | 3 | 2009-12-10 08:47:21 | 2 | Kumamoto orange L | orange | 30 | 2009-12-10 07:16:00 | |
| | 2 | 1001 | 2009 | 10 | 28 | 1 | 3 | 2009-12-10 08:47:21 | 3 | Kumamoto orange M | orange | 20 | 2009-12-10 07:16:00 | ٩. |
| ŀ | 4 | 1001 | 2009 | 10 | 28 | 1 | 3 | 2009-12-10 08:47:21 | 4 | Fukuoka melon | melon | {null} | 2009-12-10 07:16:00 | |
| | 5 | 1002 | 2009 | 11 | 1 | 2 | 1 | 2009-12-10 08:47:21 | 1 | Fukuoka apple | apple | 50 | 2009-12-10 07:16:00 | |
| | 6 | 1002 | 2009 | 11 | 1 | 2 | 1 | 2009-12-10 08:47:21 | 2 | Kumamoto orange L | orange | 30 | 2009-12-10 07:16:00 | |
| ŀ | 7 | 1002 | 2009 | 11 | 1 | 2 | 1 | 2009-12-10 08:47:21 | 3 | Kumamoto orange M | orange | 20 | 2009-12-10 07:16:00 | 11. |
| | 8 | 1002 | 2009 | 11 | 1 | 2 | 1 | 2009-12-10 08:47:21 | 4 | Fukuoka melon | melon | {null} | 2009-12-10 07:16:00 | |
| | 9 | 1003 | 2009 | 11 | 2 | 1 | 2 | 2009-12-10 08:47:21 | 1 | Fukuoka apple | apple | 50 | 2009-12-10 07:16:00 | |
| | 10 | 1003 | 2009 | 11 | 2 | 1 | 2 | 2009-12-10 08:47:21 | 2 | Kumamoto orange L | orange | 30 | 2009-12-10 07:16:00 | |
| | 11 | 1003 | 2009 | 11 | 2 | 1 | 2 | 2009-12-10 08:47:21 | 3 | Kumamoto orange M | orange | 20 | 2009-12-10 07:16:00 | 11. |
| | 12 | 1003 | 2009 | 11 | 2 | 1 | 2 | 2009-12-10 08:47:21 | 4 | Fukuoka melon | melon | {null} | 2009-12-10 07:16:00 | |
| | 13 | 1004 | 2009 | 11 | 2 | 3 | 4 | 2009-12-10 08:47:21 | 1 | Fukuoka apple | apple | 50 | 2009-12-10 07:16:00 | |
| | 14 | 1004 | 2009 | 11 | 2 | 3 | 4 | 2009-12-10 08:47:21 | 2 | Kumamoto orange L | orange | 30 | 2009-12-10 07:16:00 | |
| | 15 | 1004 | 2009 | 11 | 2 | 3 | 4 | 2009-12-10 08:47:21 | 3 | Kumamoto orange M | orange | 20 | 2009-12-10 07:16:00 | |
| | 16 | 1004 | 2009 | 11 | 2 | 3 | 4 | 2009-12-10 08:47:21 | 4 | Fukuoka melon | melon | {null} | 2009-12-10 07:16:00 | J |

結合問い合わせ (join query)

| | requests | 5 | | | | | product | 15 | | |
|---|----------------------|----------------------|----------------|-------------|-------------|-------------|---------|---|---------------------------|------|
| | id | year | month | day | product id | qty | id | product_name | type | cost |
| [| 1 001 | 2009 | 10 | 28 | 1 | 3 | 1 | Fukuoka apple | apple | 50 |
| | 1 002 | 2009 | 11 | 1 | 2 | 1 | 2 | Kumamoto orange L | orange | 30 |
| [| 1 003 | 2009 | 11 | 2 | 1 | 2 | 3 | Kumamoto orange M | orange | 20 |
| l | 1 0 0 4 | 2009 | 11 | 2 | 3 | 4 | 4 | Fukuoka melon | melon | NULL |
| ļ | 1002 1003 1004 | 2009 2009 2009 | 11 11 11 | 1 2 2 | 2 1 3 | 1 2 4 | 2 3 | Kumamoto orange L Kumamoto orange M Fukuoka melon | orange orange melon | |

結合問い合わせは、直積から、条件を満足する行を選んだものになる.

List all 'name', 'price' and 'requests.qty' that satisfy "requests.month = $10^{"}$

| SELECT products.product_name, products.cost, request FROM products, requests WHERE products.id = requests.product_id AND requests.month = 10; | s.qty | |
|---|--------------------|--------------------|
| DR II = 2 2 | 9 | |
| 1 SELECT products.product_name, products.s 2 FROM products, requests 3 WHERE products.id = requests.product_id 4 AND requests.month = 10; | cost, requests.qty | |
| | | Þ |
| uration: 0.016 seconds | | * Col: 36 Row: 4/4 |
| B B B B B B B D [] [| \$ | |
| Full View Item View Script Output | | |
| product_name | cost | qty |
| 1 Fukuoka apple | 50 | 3 |
| Y | | |

SQL を用いたテーブル定義と一貫性制約の記述 (Table definition and integrity constraint specification using SQL)



SQL を用いて, **bundles テーブル, shippings テーブルを定義し, 一貫性制約を記述**する. (Define two table 'bundles' and 'shippings'. Specify integrity constrants of the table using SQL)

1. bundles テーブルの定義

次の SQL を入力し、「Run SQL」のアイコンをクリック (Write the following SQL, and click "Run SQL" icon).

CREATE TABLE bundles (id INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL, request_id INTEGER NOT NULL REFERENCES requests(id), qty INTEGER NOT NULL, shipping_id INTEGER NOT NULL REFERENCES shippings(id), created_at DATETIME NOT NULL):



2. コンソールの確認 (Inspect console)



エラーメッセージが出ていないことを確認

3. shippings テーブルの定義

次の SQL を入力し,「Run SQL」のアイコンをクリック (Write the following SQL, and click "Run SQL" icon).

```
CREATE TABLE shippings (
id INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,
year INTEGER NOT NULL CHECK ( year > 2008 ),
month INTEGER NOT NULL CHECK ( month >= 1 AND month <= 12 ),
day INTEGER NOT NULL CHECK ( day >= 1 AND day <= 31 ),
created_at DATETIME NOT NULL );
```



4. コンソールの確認 (Inspect console)



エラーメッセージが出ていないことを確認

5. テーブル一覧の表示 (List of tables)

オブジェクト・ブラウザ (Object Browser) の中の「Tables」を展開 (Click 'Tables'). テーブルー覧が表示される.



SQL を用いたテーブルへの行の挿入 (Insert rows into a table using SQL)

| bundles | | | |
|---------|------------|-----|-------------|
| id | request id | qty | shipping id |
| 1 | 1001 | 2 | 101 |
| 2 | 1001 | 1 | 102 |
| 3 | 1002 | 1 | 103 |
| 4 | 1003 | 2 | 104 |
| 5 | 1004 | 4 | 104 |

| shipping | ş | | |
|----------|------|-------|-----|
| id | year | month | day |
| 101 | 2009 | 10 | 28 |
| 102 | 2009 | 10 | 31 |
| 103 | 2009 | 11 | 1 |
| 104 | 2009 | 11 | 2 |

テーブル間の関係

| I | shinnings | bundle | 5 | | | | maueste | | | | | | | | | | |
|---|-------------------|----------|------------|-----|-------------|--------|---------|------|-------|-----|------------|-----|---|-----|-------------------|--------|------|
| | id vear month day | <u> </u> | request id | atv | shipping id | | ä | year | month | day | product id | qty | | ы | product_name | type | cost |
| | 101 2009 10 28 | - | 1001 | 2 | 101 | \geq | 1 0 0 1 | 2009 | 10 | 28 | 1 | 3 | | 1 | Fukuoka apple | apple | 50 |
| 1 | 102 2009 10 31 | | 1001 | | 102 | _ | 1 002 | 2009 | 11 | 1 | 2 | 1 | | 2 | Kumamoto orange L | orange | 30 |
| 1 | 103 2009 11 1 | 4 | 1002 | 2 | 103 | - | 1008 | 2009 | 11 | 2 | 1 | 2 | / | - 3 | Kumamoto orange M | orange | 20 |
| | 104 2009 11 2 | 5 | 1004 | 4 | 104 | - | 1004 | 2009 | 11 | 2 | 3 | 4 | | 4 | Fukuoka melon | melon | NULL |

■ 以下の手順で, SQL を用いて bundles テーブルへの行の挿入を行う (Insert rows into table 'bundles' using SQL)

1. SQL プログラムの記述

| BEGIN TRANSACTION: | | | |
|---------------------------------|--------------|---------|--------------------|
| INSERT INTO bundles VALU | JES(1, 1001, | 2, 101, | datetime('now')); |
| INSERT INTO bundles VALU | JES(2, 1001, | 1, 102, | datetime('now')) |
| INSERT INTO bundles VALU | JES(3, 1002, | 1, 103, | datetime('now')); |
| INSERT INTO bundles VALU | JES(4, 1003, | 2, 104, | datetime('now')) |
| INSERT INTO bundles VALU | JES(5, 1004, | 4, 104, | datetime('now')) |
| COMMIT | | | |

| S mydb - Sqliteman | | |
|--|---|--|
| <u>File Context Database System H</u> elp | | |
| Sche <u>m</u> a <u>P</u> ragmas | 🗍 🕨 🦹 🖿 💼 📄 🗁 🏝 🔍 | 6 |
| Database → → main → → Tables (6) ⊕ - bundles ⊕ - order_records ⊕ - products ⊕ - requests | BEGIN TRANSACTION; INSERT INTO bundles VALUES(1, 1001, 2, 1) INSERT INTO bundles VALUES(2, 1001, 1, 1) INSERT INTO bundles VALUES(3, 1002, 1, 1) INSERT INTO bundles VALUES(3, 1003, 2, 1) INSERT INTO bundles VALUES(4, 1003, 2, 1) INSERT INTO bundles VALUES(5, 1004, 4, 1) COMMIT | 01, datetime(now')); 02, datetime(now')); 03, datetime(now')); 04, datetime(now')); 04, datetime(now')); |

2. 複数の SQL 文の一括実行 (Run multiple SQL statements)

複数の SQL 文を一括実行したいので, **カーソルを先頭行に移動**した後に,「Run multiple SQL statements ...」のボタ ンをクリックする.「Move the cursor to the top statement. Click ["]Run multiple SQL statements from current cursor position in one batch["] icon)



- 3. 「Script Output」ウインドウの確認 (Inspect "Script Output" window)
 - エラーメッセージが出ていないことを確認

| Full Vi | iew Item View Script Output | |
|---------|---|----------------------|
| - | Carried adapted | |
| 5 | | |
| 2 | No error | |
| 4 | | |
| 5 | INSERT INTO bundles VALUES(1, 1001, 2, 101 | 1 datetime(now)) |
| 6 | No error | datetime them y, |
| 7 | | |
| 8 | INSERT INTO bundles VALUES(2, 1001, 1, 102 | 2. datetime('now')); |
| 9 | No error | |
| 10 | | |
| 11 | INSERT INTO bundles VALUES(3, 1002, 1, 103 | 3, datetime(now')); |
| 12 | No error | |
| 13 | | |
| 14 | INSERT INTO bundles VALUES(4, 1003, 2, 104 | 4, datetime(now')); |
| 15 | No error | |
| 16 | | |
| 17 | INSERT INTO bundles VALUES (5, 1004, 4, 104 | 4, datetime(now')); |
| 18 | No error | |
| 19 | | |
| 20 | COMMIT | |
| 21 | No error | |
| 22 | | |
| 23 | Script finished | |

- 以下の手順で, SQL を用いて shippings テーブルへの行の挿入を行う (Insert rows into table 'shippings' using SQL)
 - 1. SQL プログラムの記述

| BEGIN | TRANS/ | ACTION: | | | | | | | |
|--------|--------|-----------|---------|------|-------|-----|-----|----------------------------|----|
| INSERT | INTO | shippings | VALUES(| 101, | 2009, | 10, | 28, | <pre>datetime('now')</pre> |); |
| INSERT | INTO | shippings | VALUES(| 102, | 2009, | 10, | 31, | <pre>datetime('now')</pre> |); |
| INSERT | INTO | shippings | VALUES(| 103, | 2009, | 11, | 1, | datetime('now') |): |
| INSERT | INTO | shippings | VALUES(| 104, | 2009, | 11, | 2, | <pre>datetime('now')</pre> |); |
| COMMIT | ; | | | | | | | | |

| S mydb - Sqliteman | |
|--|--|
| <u>File Context</u> <u>Database</u> <u>System</u> <u>H</u> elp | |
| Sche <u>m</u> a <u>P</u> ragmas | |
| Database imain imain image: Tables (6) image: Dundles image: Order_records image: products | 1 (BEGIN TRANSACTION: INSERT INTO shippings VALUES(101, 2009, 10, 28, datetime(now)); INSERT INTO shippings VALUES(102, 2009, 10, 31, datetime(now)); INSERT INTO shippings VALUES(103, 2009, 11, 1, datetime(now)); INSERT INTO shippings VALUES(104, 2009, 11, 2, datetime(now)); COMMIT; |

2. 複数の SQL 文の一括実行 (Run multiple SQL statements)

複数の SQL 文を一括実行したいので, **カーソルを先頭行に移動**した後に、「Run multiple SQL statements ...」のボタ ンをクリックする. 「Move the cursor to the top statement. Click "Run multiple SQL statements from current cursor position in one batch" icon)

| S mydb - Sqliteman | |
|---|--|
| <u>File Context Database System H</u> elp | |
| Sche <u>m</u> a <u>P</u> ragmas | |
| Database | |
| 🖻 🖳 😂 main | [2] INSERT INTO shippings VALUES(101, 2009, 10, 28, datetime(now)); INSERT INTO shippings VALUES(102, 2009, 10, 21, datetime(now)); |
| 🔁 🛄 Tables (6) | 4 INSERT INTO shippings VALUES(103, 2009, 10, 01, date(inter(inter(inter))); |
| ⊕ bundles | 5 INSERT INTO shippings VALUES(104, 2009, 11, 2, datetime(now)); |
| œrder_records | 6 COMMIT; |
| | |

3. 「Script Output」ウインドウの確認 (Inspect "Script Output" window)

エラーメッセージが出ていないことを確認

| Full V | iew I Item View Script Output | | |
|----------------|---|------------------------------|--|
| 1 2 3 | Script started BEGIN TRANSACTION; No error | | |
| 4 5 6 | INSERT INTO shippings VALUES (101, 2009 No error | , 10, 28, datetime(now')); | |
| 7 8 9 | INSERT INTO shippings VALUES (102, 2009 No error | , 10, 31, datetime(now')); | |
| 10 11 12 | INSERT INTO shippings VALUES (103, 2009 No error | . 11, 1, datetime('now')); | |
| 13 14 15 | INSERT INTO shippings VALUES (104, 2009 No error | , 11, -2, datetime('now')); | |
| 10 17 18 | COMMIT; No error | | |
| 20 21 | Script finished | | |

SQLiteman を用いたデータのブラウズ (Browse Data using SQLiteman)

・bundles テーブル

テーブル **bundles**を選ぶ (Select table 'products')

| _ | | |
|------|---|---|
| S | mydb – Sqliteman | |
| Eile | e <u>C</u> ontext <u>D</u> atabase <u>S</u> ystem | H |
| | | |
| | Sche <u>m</u> a <u>P</u> ragmas | |
| | Detabase | _ |
| | Database | |
| | 🖻 🚽 😹 main | |
| | 🖻 🎹 Tables (6) | |
| | • bundles | |
| | order_records | |
| | ⊡ products | |
| | | |
| | ⊡ score_records | |
| | | |
| | 💼 Views (0) | |
| | 🛓 🔄 🔄 System Catalogue (2) | |
| | | |

テーブル **bundles**が表示される (table 'products' appears)

| Full \ | √iew Item | View Script Output | | | |
|--------|-----------|----------------------|-----|-------------|---------------------|
| | id | request_id | qty | shipping_id | created_at |
| 1 | 1 | 1001 | 2 | 101 | 2009-12-10 09:00:11 |
| 2 | 2 | 1001 | 1 | 102 | 2009-12-10 09:00:11 |
| 3 | 3 | 1002 | 1 | 103 | 2009-12-10 09:00:11 |
| 4 | 4 | 1003 | 2 | 104 | 2009-12-10 09:00:11 |
| 5 | 5 | 1004 | 4 | 104 | 2009-12-10 09:00:11 |
| | 1 | | | | |

※ もし, データに間違いがあれば, このウインドウで修正できる (If you find any mistakes, you can modify the data using this window).

・shippings テーブル

| F | ull View | Item Vier | w Script Output | | | |
|----|----------|-----------|-------------------|-------|-----|---------------------|
| ſГ | _ | id | year | month | day | created_at |
| 1 | | 101 | 2009 | 10 | 28 | 2009-12-10 09:04:12 |
| 2 | 2 | 102 | 2009 | 10 | 31 | 2009-12-10 09:04:12 |
| 3 | 3 | 103 | 2009 | 11 | 1 | 2009-12-10 09:04:12 |
| 4 | + | 104 | 2009 | 11 | 2 | 2009-12-10 09:04:12 |

SQL 問い合わせの発行と評価結果の確認 (Issue SQL queries and inspect the results)

直積集合 (Cartesian product)

SQL を使い、複数のテーブルからの直積を簡単に得ることができる.

SELECT * FROM shippings, bundles, requests;

| 2 SELECT * FROM shippings, bundles, requests] 2 FROM shippings, bundles, requests] 2 Col: 35 Row: 2/2 2 Image: Col: 35 Row: 2/2 3 Image: Col: 35 Row: 2/2 1 101 1 101 2009 10 28 2009-12-10 090412 1 1 101 2009 10 28 2009-12-10 090412 1 3 101 2009 10 28 2009-12-10 090412 1 1001 2 101 2009-12-10 090411 1 4 101 2009 10 28 2009-12-10 090412 1 1001 2 101 2009-12-10 090411 1 5 101 2009 10 28 2009-12-10 090412 2 1001 1 102 2009-12-10 090411 1 1 101 2009 10 28 2009-12-10 090412 2 1001 1 102 2009-12-10 090011 1 | U | | | | | | | | | | | | |
|---|-----|------------------------------------|---------|---------|--------|-----|---------------------|----|------------|-----|-------------|---------------------|------------|
| Image: controles, requests Image: controles, requests Image: control contecontrol control control control control control conte | 1 | | SEL | ECT * | | | | | | | | | |
| Image: constraint of the second sec | 2 | Trom supprings, Dulhules, requests | | | | | | | | | | | |
| Image: constraint of the second sec | | | | | | | | | | | | | |
| Nuration: 0.031 seconds Coi: 35 Rov: 2/2 Image: Coi: 0.050 monomed and the second and the secon | • | | | | | | | | | | | | Þ |
| Image: Construct of the system of t | Dur | atio | n: 0.03 | 1 secor | nds | | | | | | | * Col: 35 Ro | w: 2/2 |
| Full View Item View Script Output id year month day created_at id request_id dty shipping_id created_at id 1 101 2009 10 28 2009-12-10 090412 1 1001 2 101 2009-12-10 090011 2 101 2009 10 28 2009-12-10 090412 1 1001 2 101 2009-12-10 090011 4 101 2009 10 28 2009-12-10 090412 1 1001 2 101 2009-12-10 090011 5 101 2009 10 28 2009-12-10 090412 2 1001 1 102 2009-12-10 090011 5 101 2009 10 28 2009-12-10 090412 1 101 102 2009-12-10 090011 5 1002 1 103 2009-12-10 090011 1 103 2009-12-10 09001 | E | | | | æ .: | × | | | | | | | |
| Full View Item View Script Output id year month day created_at id request_id qty shipping_id created_at id 1 101 2009 10 28 2009-12-10 09.0412 1 1001 2 101 2009-12-10 09.0011 1 2 101 2009 10 28 2009-12-10 09.0412 1 1001 2 101 2009-12-10 09.0011 1 4 101 2009 10 28 2009-12-10 09.0412 1 1001 2 101 2009-12-10 09.0011 1 5 101 2009 10 28 2009-12-10 09.0412 2 1001 1 102 2009-12-10 09.0011 1 6 101 2009 10 28 2009-12-10 9.0412 2 1001 1 102 2009-12-10 9.0011 1 8 | | - | | | | 3 | | | | | | | |
| id year month day ccreated_at id request_id qty shipping_id ccreated_at A 1 101 2009 10 28 2009-12-10 09.0412 1 1001 2 101 2009-12-10 09.0011 1 3 101 2009 10 28 2009-12-10 09.0412 1 1001 2 101 2009-12-10 09.0011 4 101 2009 10 28 2009-12-10 09.0412 1 1001 2 101 2009-12-10 09.0011 5 101 2009 10 28 2009-12-10 09.0412 2 1001 1 102 2009-12-10 09.011 1 102 2009-12-10 09.011 1 102 2009-12-10 09.011 1 102 2009-12-10 09.011 1 102 2009-12-10 09.011 1 103 2009-12-10 09.011 1 103 2009-12-10< | 1 | Full | View | Iten | n View | Scr | ipt Output | | | | | | |
| 1 101 209 10 28 2009-12-10 090412 1 1001 2 101 2009-12-10 090011 1 2 101 2009 10 28 2009-12-10 090412 1 1001 2 101 2009-12-10 090011 1 3 101 2009 10 28 2009-12-10 090412 1 1001 2 101 2009-12-10 090011 4 101 2009 10 28 2009-12-10 090412 2 1001 1 102 2009-12-10 090011 5 101 2009 10 28 2009-12-10 090412 2 1001 1 102 2009-12-10 090011 1 102 2009-12-10 090011 1 102 2009-12-10 090011 1 102 2009-12-10 090011 1 103 2009-12-10 090011 1 103 2009-12-10 090011 1 103 2009-12-10 090011 1 103 2009-12-10 090011 1 | ſ | | id | year | month | day | created_at | id | request_id | qty | shipping_id | created_at | |
| 2 101 209 10 28 2009-12-10 090412 1 1001 2 101 2009-12-10 090011 3 101 2009 10 28 2009-12-10 090412 1 1001 2 101 2009-12-10 090011 4 101 2009 10 28 2009-12-10 090412 2 1001 1 102 2009-12-10 090011 5 101 2009 10 28 2009-12-10 090412 2 1001 1 102 2009-12-10 090011 6 101 2009 10 28 2009-12-10 090412 2 1001 1 102 2009-12-10 090011 7 101 2009 10 28 2009-12-10 090412 2 1001 1 102 2009-12-10 090011 8 101 2009 10 28 2009-12-10 090412 3 1002 1 103 2009-12-10 090011 1 10 101 2 | | 1 | 101 | 2009 | 10 | 28 | 2009-12-10 09:04:12 | 1 | 1001 | 2 | 101 | 2009-12-10 09:00:11 | |
| 3 101 2009 10 28 2009-12-10 909.112 1 1001 2 101 2009-12-10 090.011 1 4 101 2009 10 28 2009-12-10 090.412 1 1001 2 101 2009-12-10 090.011 1 5 101 2009 10 28 2009-12-10 090.412 2 1001 1 102 2009-12-10 090.011 1 6 101 2009 10 28 2009-12-10 090.412 2 1001 1 102 2009-12-10 090.011 1 102 2009-12-10 090.011 1 102 2009-12-10 090.011 1 102 2009-12-10 090.011 1 103 2009-12-10 090.011 1 103 2009-12-10 090.011 1 103 2009-12-10 090.011 1 103 2009-12-10 090.011 1 103 2009-12-10 090.011 1 103 2009-12-10 090.011 1 11 101 2009 10 < | | 2 | 101 | 2009 | 10 | 28 | 2009-12-10 09:04:12 | 1 | 1001 | 2 | 101 | 2009-12-10 09:00:11 | |
| 4 101 2009 10 28 2009-12-10 090412 1 1001 2 101 2009-12-10 090011 1 5 101 2009 10 28 2009-12-10 090412 2 1001 1 102 2009-12-10 090011 1 6 101 2009 10 28 2009-12-10 090412 2 1001 1 102 2009-12-10 090011 1 7 101 2009 10 28 2009-12-10 090412 2 1001 1 102 2009-12-10 090011 1 8 101 2009 10 28 2009-12-10 090412 3 1002 1 103 2009-12-10 090011 1 10 101 2009 10 28 2009-12-10 090412 3 1002 1 103 2009-12-10 090011 1 103 2009-12-10 090011 1 113 2009-12-10 090011 1 103 2009-12-10 090011 1 <td< td=""><td></td><td>3</td><td>101</td><td>2009</td><td>10</td><td>28</td><td>2009-12-10 09:04:12</td><td>1</td><td>1001</td><td>2</td><td>101</td><td>2009-12-10 09:00:11</td><td></td></td<> | | 3 | 101 | 2009 | 10 | 28 | 2009-12-10 09:04:12 | 1 | 1001 | 2 | 101 | 2009-12-10 09:00:11 | |
| 5 101 2009 10 28 2009-12-10 090412 2 1001 1 102 2009-12-10 090011 1 6 101 2009 10 28 2009-12-10 090412 2 1001 1 102 2009-12-10 090011 1 7 101 2009 10 28 2009-12-10 090412 2 1001 1 102 2009-12-10 090011 1 8 101 2009 10 28 2009-12-10 090412 3 1002 1 103 2009-12-10 090011 1 9 101 2009 10 28 2009-12-10 090412 3 1002 1 103 2009-12-10 090011 1 10 101 2009 10 28 2009-12-10 090412 3 1002 1 103 2009-12-10 090011 1 12 101 2009 10 28 2009-12-10 090412 4 1003 2 104 2009-12-10 | | 4 | 101 | 2009 | 10 | 28 | 2009-12-10 09:04:12 | 1 | 1001 | 2 | 101 | 2009-12-10 09:00:11 | |
| 6 101 2009 10 28 2009-12-10 0904:12 2 1001 1 102 2009-12-10 090.011 1 7 101 2009 10 28 2009-12-10 0904:12 2 1001 1 102 2009-12-10 090.011 1 8 101 2009 10 28 2009-12-10 0904:12 3 1002 1 103 2009-12-10 090.011 1 9 101 2009 10 28 2009-12-10 0904:12 3 1002 1 103 2009-12-10 090.011 1 10 101 2009 10 28 2009-12-10 09.04:12 3 1002 1 103 2009-12-10 09.00:11 1 12 101 2009 10 28 2009-12-10 09.04:12 3 1002 1 103 2009-12-10 09.00:11 1 13 101 2009 10 28 2009-12-10 09.04:12 4 1003 2 104 <td< td=""><td></td><td>5</td><td>101</td><td>2009</td><td>10</td><td>28</td><td>2009-12-10 09:04:12</td><td>2</td><td>1001</td><td>1</td><td>102</td><td>2009-12-10 09:00:11</td><td></td></td<> | | 5 | 101 | 2009 | 10 | 28 | 2009-12-10 09:04:12 | 2 | 1001 | 1 | 102 | 2009-12-10 09:00:11 | |
| 7 101 2009 10 28 2009-12-10 09.04:12 2 1001 1 102 2009-12-10 09.00:11 1 9 101 2009 10 28 2009-12-10 09.04:12 3 1002 1 103 2009-12-10 09.00:11 1 9 101 2009 10 28 2009-12-10 09.04:12 3 1002 1 103 2009-12-10 09.00:11 1 10 101 2009 10 28 2009-12-10 09.04:12 3 1002 1 103 2009-12-10 09.00:11 1 11 101 2009 10 28 2009-12-10 09.04:12 3 1002 1 103 2009-12-10 09.00:11 1 12 101 2009 10 28 2009-12-10 09.04:12 4 1003 2 104 2009-12-10 09.00:11 1 13 101 2009 10 28 2009-12-10 09.04:12 4 1003 2 104 | | 6 | 101 | 2009 | 10 | 28 | 2009-12-10 09:04:12 | 2 | 1001 | 1 | 102 | 2009-12-10 09:00:11 | - |
| 8 101 2009 10 28 2009-12-10 09.04:12 2 1001 1 102 2009-12-10 09.00:11 1 9 101 2009 10 28 2009-12-10 09.04:12 3 1002 1 103 2009-12-10 09.00:11 1 10 101 2009 10 28 2009-12-10 09.04:12 3 1002 1 103 2009-12-10 09.00:11 1 11 101 2009 10 28 2009-12-10 09.04:12 3 1002 1 103 2009-12-10 09.00:11 1 12 101 2009 10 28 2009-12-10 09.04:12 4 1003 2 104 2009-12-10 09.00:11 1 13 101 2009 10 28 2009-12-10 09.04:12 4 1003 2 104 2009-12-10 09.00:11 1 14 101 2009 10 28 2009-12-10 09.04:12 5 1004 4 104 | | 7 | 101 | 2009 | 10 | 28 | 2009-12-10 09:04:12 | 2 | 1001 | 1 | 102 | 2009-12-10 09:00:11 | |
| 9 101 2009 10 28 2009-12-10 09.04:12 3 1002 1 103 2009-12-10 09.00:11 1 10 101 2009 10 28 2009-12-10 09.04:12 3 1002 1 103 2009-12-10 09.00:11 1 11 101 2009 10 28 2009-12-10 09.04:12 3 1002 1 103 2009-12-10 09.00:11 1 12 101 2009 10 28 2009-12-10 09.04:12 3 1002 1 103 2009-12-10 09.00:11 1 13 101 2009 10 28 2009-12-10 09.04:12 4 1003 2 104 2009-12-10 09.00:11 1 14 101 2009 10 28 2009-12-10 09.04:12 4 1003 2 104 2009-12-10 09.00:11 1 15 101 2009 10 28 2009-12-10 09.04:12 5 1004 4 104 <td></td> <td>8</td> <td>101</td> <td>2009</td> <td>10</td> <td>28</td> <td>2009-12-10 09:04:12</td> <td>2</td> <td>1001</td> <td>1</td> <td>102</td> <td>2009-12-10 09:00:11</td> <td>†.</td> | | 8 | 101 | 2009 | 10 | 28 | 2009-12-10 09:04:12 | 2 | 1001 | 1 | 102 | 2009-12-10 09:00:11 | † . |
| 10 101 2009 10 28 2009-12-10 09.04:12 3 1002 1 103 2009-12-10 09.00:11 1 11 101 2009 10 28 2009-12-10 09.04:12 3 1002 1 103 2009-12-10 09.00:11 1 12 101 2009 10 28 2009-12-10 09.04:12 3 1002 1 103 2009-12-10 09.00:11 1 13 101 2009 10 28 2009-12-10 09.04:12 4 1003 2 104 2009-12-10 09.00:11 1 14 101 2009 10 28 2009-12-10 09.04:12 4 1003 2 104 2009-12-10 09.00:11 1 15 101 2009 10 28 2009-12-10 09.04:12 4 1003 2 104 2009-12-10 09.00:11 1 16 101 2009 10 28 2009-12-10 09.04:12 5 1004 4 104 <td></td> <td>9</td> <td>101</td> <td>2009</td> <td>10</td> <td>28</td> <td>2009-12-10 09:04:12</td> <td>3</td> <td>1002</td> <td>1</td> <td>103</td> <td>2009-12-10 09:00:11</td> <td></td> | | 9 | 101 | 2009 | 10 | 28 | 2009-12-10 09:04:12 | 3 | 1002 | 1 | 103 | 2009-12-10 09:00:11 | |
| 11 101 2009 10 28 2009-12-10 09.04:12 3 1002 1 103 2009-12-10 09.00:11 1 12 101 2009 10 28 2009-12-10 09.04:12 3 1002 1 103 2009-12-10 09.00:11 1 13 101 2009 10 28 2009-12-10 09.04:12 4 1003 2 104 2009-12-10 09.00:11 1 14 101 2009 10 28 2009-12-10 09.04:12 4 1003 2 104 2009-12-10 09.00:11 1 15 101 2009 10 28 2009-12-10 09.04:12 4 1003 2 104 2009-12-10 09.00:11 1 16 101 2009 10 28 2009-12-10 09.04:12 5 1004 4 104 2009-12-10 09.00:11 1 17 101 2009 10 28 2009-12-10 09.04:12 5 1004 4 104 <td></td> <td>10</td> <td>101</td> <td>2009</td> <td>10</td> <td>28</td> <td>2009-12-10 09:04:12</td> <td>3</td> <td>1002</td> <td>1</td> <td>103</td> <td>2009-12-10 09:00:11</td> <td>†.</td> | | 10 | 101 | 2009 | 10 | 28 | 2009-12-10 09:04:12 | 3 | 1002 | 1 | 103 | 2009-12-10 09:00:11 | † . |
| 12 101 2009 10 28 2009-12-10 09.04:12 3 1002 1 103 2009-12-10 09.00:11 1 13 101 2009 10 28 2009-12-10 09.04:12 4 1003 2 104 2009-12-10 09.00:11 1 14 101 2009 10 28 2009-12-10 09.04:12 4 1003 2 104 2009-12-10 09.00:11 1 15 101 2009 10 28 2009-12-10 09.04:12 4 1003 2 104 2009-12-10 09.00:11 1 16 101 2009 10 28 2009-12-10 09.04:12 4 1003 2 104 2009-12-10 09.00:11 1 16 101 2009 10 28 2009-12-10 09.04:12 5 1004 4 104 2009-12-10 09.00:11 1 18 101 2009 10 28 2009-12-10 09.04:12 5 1004 4 104 <td></td> <td>11</td> <td>101</td> <td>2009</td> <td>10</td> <td>28</td> <td>2009-12-10 09:04:12</td> <td>3</td> <td>1002</td> <td>1</td> <td>103</td> <td>2009-12-10 09:00:11</td> <td></td> | | 11 | 101 | 2009 | 10 | 28 | 2009-12-10 09:04:12 | 3 | 1002 | 1 | 103 | 2009-12-10 09:00:11 | |
| 13 101 2009 10 28 2009-12-10 09.04:12 4 1003 2 104 2009-12-10 09.00:11 1 14 101 2009 10 28 2009-12-10 09.04:12 4 1003 2 104 2009-12-10 09.00:11 1 15 101 2009 10 28 2009-12-10 09.04:12 4 1003 2 104 2009-12-10 09.00:11 1 16 101 2009 10 28 2009-12-10 09.04:12 4 1003 2 104 2009-12-10 09.00:11 1 16 101 2009 10 28 2009-12-10 09.04:12 5 1004 4 104 2009-12-10 09.00:11 1 17 101 2009 10 28 2009-12-10 09.04:12 5 1004 4 104 2009-12-10 09.00:11 1 19 101 2009 10 28 2009-12-10 09.04:12 5 1004 4 104 <td></td> <td>12</td> <td>101</td> <td>2009</td> <td>10</td> <td>28</td> <td>2009-12-10 09:04:12</td> <td>3</td> <td>1002</td> <td>1</td> <td>103</td> <td>2009-12-10 09:00:11</td> <td>T. </td> | | 12 | 101 | 2009 | 10 | 28 | 2009-12-10 09:04:12 | 3 | 1002 | 1 | 103 | 2009-12-10 09:00:11 | T. |
| 14 101 2009 10 28 2009-12-10 09:04:12 4 1003 2 104 2009-12-10 09:00:11 1 15 101 2009 10 28 2009-12-10 09:04:12 4 1003 2 104 2009-12-10 09:00:11 1 16 101 2009 10 28 2009-12-10 09:04:12 4 1003 2 104 2009-12-10 09:00:11 1 16 101 2009 10 28 2009-12-10 09:04:12 5 1004 4 104 2009-12-10 09:00:11 1 17 101 2009 10 28 2009-12-10 09:04:12 5 1004 4 104 2009-12-10 09:00:11 1 18 101 2009 10 28 2009-12-10 09:04:12 5 1004 4 104 2009-12-10 09:00:11 1 20 101 2009 10 31 2009-12-10 09:04:12 1 1001 2 101 <td></td> <td>13</td> <td>101</td> <td>2009</td> <td>10</td> <td>28</td> <td>2009-12-10 09:04:12</td> <td>4</td> <td>1003</td> <td>2</td> <td>104</td> <td>2009-12-10 09:00:11</td> <td></td> | | 13 | 101 | 2009 | 10 | 28 | 2009-12-10 09:04:12 | 4 | 1003 | 2 | 104 | 2009-12-10 09:00:11 | |
| 15 101 2009 10 28 2009-12-10 09.04:12 4 1003 2 104 2009-12-10 09.00:11 1 16 101 2009 10 28 2009-12-10 09.04:12 4 1003 2 104 2009-12-10 09.00:11 1 17 101 2009 10 28 2009-12-10 09.04:12 5 1004 4 104 2009-12-10 09.00:11 1 18 101 2009 10 28 2009-12-10 09.04:12 5 1004 4 104 2009-12-10 09.00:11 1 18 101 2009 10 28 2009-12-10 09.04:12 5 1004 4 104 2009-12-10 09.00:11 1 19 101 2009 10 28 2009-12-10 09.04:12 5 1004 4 104 2009-12-10 09.00:11 1 20 102 2009 10 31 2009-12-10 09.04:12 1 1001 2 101 <td></td> <td>14</td> <td>101</td> <td>2009</td> <td>10</td> <td>28</td> <td>2009-12-10 09:04:12</td> <td>4</td> <td>1003</td> <td>2</td> <td>104</td> <td>2009-12-10 09:00:11</td> <td>†</td> | | 14 | 101 | 2009 | 10 | 28 | 2009-12-10 09:04:12 | 4 | 1003 | 2 | 104 | 2009-12-10 09:00:11 | † |
| 16 101 2009 10 28 2009-12-10 09:04:12 4 1003 2 104 2009-12-10 09:00:11 1 17 101 2009 10 28 2009-12-10 09:04:12 5 1004 4 104 2009-12-10 09:00:11 1 18 101 2009 10 28 2009-12-10 09:04:12 5 1004 4 104 2009-12-10 09:00:11 1 19 101 2009 10 28 2009-12-10 09:04:12 5 1004 4 104 2009-12-10 09:00:11 1 19 101 2009 10 28 2009-12-10 09:04:12 5 1004 4 104 2009-12-10 09:00:11 1 20 101 2009 10 31 2009-12-10 09:04:12 1 1001 2 101 2009-12-10 09:00:11 1 21 102 2009 10 31 2009-12-10 09:04:12 1 1001 2 101 <td></td> <td>15</td> <td>101</td> <td>2009</td> <td>10</td> <td>28</td> <td>2009-12-10 09:04:12</td> <td>4</td> <td>1003</td> <td>2</td> <td>104</td> <td>2009-12-10 09:00:11</td> <td></td> | | 15 | 101 | 2009 | 10 | 28 | 2009-12-10 09:04:12 | 4 | 1003 | 2 | 104 | 2009-12-10 09:00:11 | |
| 17 101 2009 10 28 2009-12-10 09:04:12 5 1004 4 104 2009-12-10 09:00:11 1 18 101 2009 10 28 2009-12-10 09:04:12 5 1004 4 104 2009-12-10 09:00:11 1 19 101 2009 10 28 2009-12-10 09:04:12 5 1004 4 104 2009-12-10 09:00:11 1 20 101 2009 10 28 2009-12-10 09:04:12 5 1004 4 104 2009-12-10 09:00:11 1 20 101 2009 10 28 2009-12-10 09:04:12 5 1004 4 104 2009-12-10 09:00:11 1 21 102 2009 10 31 2009-12-10 09:04:12 1 1001 2 101 2009-12-10 09:00:11 1 23 102 2009 10 31 2009-12-10 09:04:12 1 1001 2 101 <td></td> <td>16</td> <td>101</td> <td>2009</td> <td>10</td> <td>28</td> <td>2009-12-10 09:04:12</td> <td>4</td> <td>1003</td> <td>2</td> <td>104</td> <td>2009-12-10 09:00:11</td> <td>T. </td> | | 16 | 101 | 2009 | 10 | 28 | 2009-12-10 09:04:12 | 4 | 1003 | 2 | 104 | 2009-12-10 09:00:11 | T. |
| 18 101 2009 10 28 2009-12-10 09:04:12 5 1004 4 104 2009-12-10 09:00:11 1 19 101 2009 10 28 2009-12-10 09:04:12 5 1004 4 104 2009-12-10 09:00:11 1 20 101 2009 10 28 2009-12-10 09:04:12 5 1004 4 104 2009-12-10 09:00:11 1 20 101 2009 10 28 2009-12-10 09:04:12 5 1004 4 104 2009-12-10 09:00:11 1 21 102 2009 10 31 2009-12-10 09:04:12 1 1001 2 101 2009-12-10 09:00:11 1 22 102 2009 10 31 2009-12-10 09:04:12 1 1001 2 101 2009-12-10 09:00:11 1 23 102 2009 10 31 2009-12-10 09:04:12 1 1001 2 101 <td></td> <td>17</td> <td>101</td> <td>2009</td> <td>10</td> <td>28</td> <td>2009-12-10 09:04:12</td> <td>5</td> <td>1004</td> <td>4</td> <td>104</td> <td>2009-12-10 09:00:11</td> <td>—</td> | | 17 | 101 | 2009 | 10 | 28 | 2009-12-10 09:04:12 | 5 | 1004 | 4 | 104 | 2009-12-10 09:00:11 | — |
| 19 101 2009 10 28 2009-12-10 09:04:12 5 1004 4 104 2009-12-10 09:00:11 1 20 101 2009 10 28 2009-12-10 09:04:12 5 1004 4 104 2009-12-10 09:00:11 1 21 102 2009 10 31 2009-12-10 09:04:12 1 1001 2 101 2009-12-10 09:00:11 1 22 102 2009 10 31 2009-12-10 09:04:12 1 1001 2 101 2009-12-10 09:00:11 1 23 102 2009 10 31 2009-12-10 09:04:12 1 1001 2 101 2009-12-10 09:00:11 1 23 102 2009 10 31 2009-12-10 09:04:12 1 1001 2 101 2009-12-10 09:00:11 1 24 102 2009 10 31 2009-12-10 09:00:11 1 1 1001 2 | | 18 | 101 | 2009 | 10 | 28 | 2009-12-10 09:04:12 | 5 | 1004 | 4 | 104 | 2009-12-10 09:00:11 | T. |
| 20 101 2009 10 28 2009-12-10 09:04:12 5 1004 4 104 2009-12-10 09:00:11 `` 21 102 2009 10 31 2009-12-10 09:04:12 1 1001 2 101 2009-12-10 09:00:11 `` 22 102 2009 10 31 2009-12-10 09:04:12 1 1001 2 101 2009-12-10 09:00:11 `` 23 102 2009 10 31 2009-12-10 09:04:12 1 1001 2 101 2009-12-10 09:00:11 `` 23 102 2009 10 31 2009-12-10 09:04:12 1 1001 2 101 2009-12-10 09:00:11 `` 24 102 2009 10 31 2009-12-10 09:00:11 `` `` 4 | | 19 | 101 | 2009 | 10 | 28 | 2009-12-10 09:04:12 | 5 | 1004 | 4 | 104 | 2009-12-10 09:00:11 | |
| 21 102 2009 10 31 2009-12-10 09:04:12 1 1001 2 101 2009-12-10 09:00:11 1 22 102 2009 10 31 2009-12-10 09:04:12 1 1001 2 101 2009-12-10 09:00:11 1 23 102 2009 10 31 2009-12-10 09:04:12 1 1001 2 101 2009-12-10 09:00:11 1 23 102 2009 10 31 2009-12-10 09:04:12 1 1001 2 101 2009-12-10 09:00:11 1 24 102 2009 10 31 2009-12-10 09:00:11 1 1 1001 2 101 2009-12-10 09:00:11 < | | 20 | 101 | 2009 | 10 | 28 | 2009-12-10 09:04:12 | 5 | 1004 | 4 | 104 | 2009-12-10 09:00:11 | - |
| 22 102 2009 10 31 2009-12-10 09:04:12 1 1001 2 101 2009-12-10 09:00:11 1 23 102 2009 10 31 2009-12-10 09:04:12 1 1001 2 101 2009-12-10 09:00:11 1 24 102 2009 10 31 2009-12-10 09:04:12 1 1001 2 101 2009-12-10 09:00:11 1 24 102 2009 10 31 2009-12-10 09:04:12 1 1001 2 101 2009-12-10 09:00:11 1 24 102 2009 10 31 2009-12-10 09:00:11 1 1 1001 2 101 2009-12-10 09:00:11 < | | 21 | 102 | 2009 | 10 | 31 | 2009-12-10 09:04:12 | 1 | 1001 | 2 | 101 | 2009-12-10 09:00:11 | |
| 23 102 2009 10 31 2009-12-10 09:04:12 1 1001 2 101 2009-12-10 09:00:11 1 24 102 2009 10 31 2009-12-10 09:04:12 1 1001 2 101 2009-12-10 09:00:11 1 24 102 2009 10 31 2009-12-10 09:04:12 1 1001 2 101 2009-12-10 09:00:11 1 | | 22 | 102 | 2009 | 10 | 31 | 2009-12-10 09:04:12 | 1 | 1001 | 2 | 101 | 2009-12-10 09:00:11 | T. |
| 24 102 2009 10 31 2009-12-10 09:04:12 1 1001 2 101 2009-12-10 09:00:11 Image: Second se | | 23 | 102 | 2009 | 10 | 31 | 2009-12-10 09:04:12 | 1 | 1001 | 2 | 101 | 2009-12-10 09:00:11 | |
| | | 24 | 102 | 2009 | 10 | 31 | 2009-12-10 09:04:12 | 1 | 1001 | 2 | 101 | 2009-12-10 09:00:11 | · - |
| | | • | | | | | | | | | | • | |

Query OK Row(s) returned: 80 SELECT * FROM shippings, bundles, requests;

結合問い合わせ (join query)

結合問い合わせは、直積から、条件を満足する行を選んだものになる.

List all 'shippings.month', 'shiping.day' and 'bundles.qty' that satisfy "requests.month = 11"

演習問題と解答例

次の問いに答えよ. その後, 下記の解答例を確認せよ. Answer the following questions. Then, inspect answers described below.

×

問い (Questions)

1. 次の SQL 問い合わせの評価結果は何か? (What is the evaluation result of the following SQL query).

SELECT products.product_name, requests.qty FROM products, requests WHERE products.id = requests.product_id AND requests.qty > 2;

2. 次の SQL 問い合わせの評価結果は何か? (What is the evaluation result of the following SQL query).

SELECT shippings.year, shippings.month, shippings.day FROM shippings, bundles, requests, products WHERE products.id = requests.product_id AND requests.id = bundles.request_id AND shippings.id = bundles.shipping_id AND cost > 20;

- 3. まず SQLite を使い, 下記のテーブルを定義しなさい (Define a table below using SQL) loan(id, branchname, amount) borrow(id, customername, loanid)
- 4. 次に SQLite を使い,下記の SQL を評価させなさい (Evaluate the following SQL) BEGIN TRANSACTION; INSERT INTO loan values(1, 'fukuoka', 1000); INSERT INTO loan values(2, 'saga', 2000); INSERT INTO loan values(3, 'saga', 1500); INSERT INTO loan values(4, 'kumamoto', 3000); INSERT INTO loan values(5, 'fukuoka', 2500); COMMIT;
- 5. 次に SQLite を使い、下記の SQL を評価させなさい (Evaluate the following SQL)

| (1) | |
|------|---|
| (2) | select * from loan; |
| (2) | select branchname from loan: |
| (3) | select distinct branchname from loan: |
| (4) | select id. branchname. amount * 1000 from loan: |
| (5) | |
| | select id from loan where branchname = 'fukuoka'; select id from loan where branchname = 'saga'; select id from loan where branchname = 'saga'; |
| (6) | select in from foan where branchname – Rumamoro , |
| (7) | select id from loan where amount < 2000 and amount >= 1000: |
| (0) | BEGIN TRANSACTION: INSERT INTO borrow values(1001, 'X', 1); INSERT INTO borrow values(1002, 'X', 2); INSERT INTO borrow values(1003, 'X', 3); INSERT INTO borrow values(1004, 'Y', 4); INSERT INTO borrow values(1005, 'X', 5); COMMIT; |
| (8) | select * |
| (9) | from Ioan, borrow |
| (10) | select customername, amount from loan, borrow: |
| (10) | select customername, amount from Ioan, borrow where Ioan.id = borrow.loanid; |
| | select customername, amount from Ioan, borrow where Ioan.id = borrow.loanid AND borrow.customername = 'X'; |

解答例 (Answers)

※ 問い合わせ結果は1つのテーブルになる. その属性名には, 元のテーブル名と属性名をドットでつなげたドット記法を用いている.

1.

| products.product_name | requests qty |
|-----------------------|--------------|
| Fukuoka apple | 3 |
| Kumamoto orange M | 4 |
| | |

2.

| shippings.year | shippings.month | shippings.day |
|----------------|-----------------|---------------|
| 2009 | 10 | 28 |
| 2009 | 10 | 31 |
| 2009 | 11 | 1 |
| 2009 | 11 | 2 |