<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) -> トランザクション

- [<u>サイトマップ</u>へ] [<u>全文検索</u>へ] [<u>統計情報</u>へ]

トランザクション



トランザクション





トランザクション









演習

演習を行うために必要になる機能や文法

BEGIN TRANSACTION, COMMIT, ROLLBACK

SQL 挿入文 (insert statement) などでのデータベース更新を行うときは、最初に「BEGIN TRANSACTION;」を実行する. データベー ス更新が終わったら「COMMIT;」または「ROLLBACK;」を実行する.

- · COMMIT: ····「BEGIN TRANSACTION;」以降の全てのデータベース更新操作を**確定**したいとき
- · ROLLBACK: ····「BEGIN TRANSACTION:」以降の全てのデータベース更新操作を破棄したいとき

DELETE FROM <table-name> WHERE <expression>;

条件に合致する行の削除

```
UPDATE <table-name> SET <attribute-name>=<expression> WHERE <expression>
```

条件に合致する行に関するデータの更新

· now

現在の日時を取得する SQLite3 の関数

A SQLite3 function to get the current datetime.

datetime

日時のデータを「YYYY-MM-DD HH:MM:SS」形式の文字列に変換する SQLite3 の関数

A SQLite3 function to convert a datetime data into the string like "YYYY-MM-DD HH:MM:SS".

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. $[File] \rightarrow [Open]$

Sqliteman Sqliteman	
<u>File Context D</u> atabase <u>S</u> ystem <u>H</u> elp	
New Ctrl+N Open Ctrl+O Recent Databases Preferences Preferences Ctrl+O	
	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			l
🖻 🖵 home 🗟 win	dowslike SQLite		
場所(<u>P</u>)	名前	•	サイズ 最終変更日
🔍 検索	📄 mydb		0 パイト 17:18
⑧ 最近開いたファイル	🗇 sqlite3_analyzer		1.7 MB 2010年08月24日
kindowslike			
🔟 デスクトップ			
ニー ファイル・システム			
- フロッピー・ドライブ			
■ ドキュメント			
■ 画像			
■ ビデオ			
🗟 ダウンロード			
** 追加(<u>A</u>) 削除(<u>R</u>)			SQLITE database
		3 ++	・ンセル(C)
		目 人 \	

■ 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>
ファイルの場所の	🔁 SQLite]-
は 最近使ったファイル で デスクトップ マイドキュメント マイ ニンピュータ マイ ネットワーク	kensutf8.sq mydb shichosonsutf8. sqlite3.exe sqlite3_analyzer SQLite Databas	sql :exe :e Browser.exe		
	ファイル名(N): ファイルの種類(<u>T</u>):	mydb SQLite database (*)	•	開(回) 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)
 - Some Pragmas

 Database

 Database

·「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')



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, **E テーブルを定義し, 一貫性制約を記述**する.(Define table 'E' and specify integrity constrants of the table using SQL)

リレーショナル・スキーマ (relational schema): E(name, score, student_name)

1. E 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 E (name TEXT NOT NULL, score INTEGER NOT NULL CHECK (score >= 0 AND score <=100), student_name TEXT NOT NULL, created_at DATETIME NOT NULL, updated_at DATETIME, UNIQUE (name, student_name));

S	mydb - Sqliteman		
<u>F</u> il	e <u>C</u> ontext <u>D</u> atabase	<u>S</u> ystem <u>H</u> elp	
	Sche <u>m</u> a <u>P</u> ragmas	[•] 😰 🖿 🔄 🗁 🖄 🔍	
	Database		
		2 name TEXT NOT NULL	
	🖃 🗃 main	3 score INTEGER NOT NULL CHECK (score >= 0 AND score <=100),	
		4 student_name TEXT NOT NULL,	
		6 undated at DATETIME NOT NULL,	
	🛨 🗤 ayst	7 UNIQUE (name, student_name));	

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

	_
Query OK	
Row(s) returned: 0 (More rows can be fetched. Scroll the resultset for more rows and/or read the documentation.)	
CREATE TABLE E (
name TEXT NOT NULL,	
score INTEGER NOT NULL CHECK (score >= 0 AND score <=100),	
student_name TEXT NOT NULL,	
created_at DATETIME NOT NULL,	
updated_at DATETIME,	
UNIQUE (name, student_name));	

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

テーブルの 'created_at' 属性には、行を挿入する日時を記録する

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

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 E VALUES(Databas	se 80,	KK ,	datetime('now'),	NULL):
INSERT INTO E VALUES(Databas	se 95	AA	datetime('now'),	NULL):
INSERT INTO E VALUES(Databas	se, 80,	'LL',	datetime('now'),	NULL);
INSERT INTO E VALUES('Program	nming', 85,	KK	datetime('now'),	NULL);
INSERT INTO E VALUES('Program	nming', 75,	'LL',	datetime('now'),	NULL);
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	
Ei	e <u>C</u> ontext <u>D</u> atabase <u>S</u> ys	tem <u>H</u> elp
	Sche <u>m</u> a <u>P</u> ragmas	
	Database	BEGIN TRANSACTION
	rin 🗋 main	2 INSERT INTO E VALUES('Database', 80, (KK', datetime(now'), NULL);
	🖃 🛢 main	3 INSERT INTO E VALUES ('Database', 95, 'AA', datetime (now'), NULL);
	🕀 💷 Tables (13)	4 INSERT INTO E VALUES ('Database' 80 (11' datetime (now') NULL)
	- Views (II)	5 INSERT INTO E VALUES ('Programming' 85 'KK', datetime(now') NULL.)
	En Sustan Ostar	6 INSERT INTO E VALUES ('Programming' 25 'IL' datatime(now)) NULL)
	🖽 📷 System Cat…	2 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
	No error
4	
5	INSERT INTO E VALUES('Database', 80, KK', datetime(now'), NULL);
6	No error
	TINSERT INTO E MALLIES ('Detabase' OF (0.0' detations (new!) NULL):
	No error
10	
11	INSERT INTO E VALUES('Database', 80, LL', datetime(now'), NULL);
12	No error
13	TINSERT INTO E VALUES ('Programming' 95 'KK' datatime (now') NULL):
15	No error
16	
17	INSERT INTO E VALUES ('Programming', 75, 'LL', datetime (now'), NULL);
18	No error
20	COMMIT
21	No error
22	
23	Script finished

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

SQLを用いた問い合わせの実行例を示す.

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

SELECT * FROM E:

Im Im							
Durati	ion: 0 seconds	Script Output	2 🛛 🖦				
F	name	score	student name	created at	updated at		
1	Database	80	КК	2009-12-13 23:06:43	{null}		
2	Database	95	AA	2009-12-13 23:06:43	{null}		
3	Database	80	LL	2009-12-13 23:06:44	{null}		
4	Programming	85	КК	2009-12-13 23:06:44	{null}		
5	Programming	75	LL	2009-12-13 23:06:44	{null}		
Ψ							

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

SELECT	*	FROM	F	WHERE	name	=	'Database'	
JLLLUI	Ŧ	L V M	ь.	WILLILL	Hame	_	Dalabase	

S mydb - Sqliteman							
<u>File Context</u> <u>D</u> atabase <u>S</u>	jystem <u>H</u> elp						
Sche <u>m</u> a <u>P</u> ragmas		• 📑 🕻	🖹 🏝 😫	6			
Database main min Tabl··· Vie··· Syst···	1 SELECT * F	ROM E WHE	RE name = 'Databas		ľ		
	Duration: 0.016 seconds * Col: 41 Row: 1/1						
			🔊 📝 🛛 👼				
	Full View Item V	/iew Scrip	t Output]		
	name	score	student_name	created_at	updated_at		
	1 Database	95	AA	2009-12-14 05:50:11	{null}		
	2 Database	80	кк	2009-12-14 05:50:11	{null}		
	3 Database	80	LL	2009-12-14 05:50:11	{null}		

SELECT score FROM E WHERE name = 'Database';

🥌 mydb - Sqliteman		2
<u>File Context</u> <u>D</u> atabase	<u>System</u> <u>H</u> elp	
Schema Pragmas		
Database ⊡ ·	SELECT score FROM E WHERE name = 'Database';	
	Duration: O seconds	* Col: 45 Row: 1/1
	Full View Item View Script Output	
]
	score	
	1	90
	2	80
	3	80

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

・E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')



※ 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).

トランザクション開始後に、トランザクション開始できない (Can not start a transaction within a transaction



トランザクション

	🕨 😰 🍽 📼 🖾 🏝 🔍								
1	1 SELECT * FROM JIGYOSYO WHERE a3 IS NULL;								
•									
Duratio	uration: 0.146 seconds								
B 2 :									
Full View Item View Script Output									
	a0	a1	a2	a3	a4	a5	a6	a7	a8
1	1101	60	608621	{null}	{null}	{null}	北海道	札幌市中央区	北一条西6丁目1-2アーバンネット札幌
2	1101	60	608640	{null}	{null}	{null}	北海道	札幌市中央区	南二条西5丁目10番2号南2西5ビル
3	1101	60	608602	{null}	{null}	{null}	北海道	札幌市中央区	北二条西1丁目1番地
4	1101	60	608554	{null}	{null}	{null}	北海道	札幌市中央区	北四条西4丁目1番地
5	1101	60	608547	{null}	{null}	{null}	北海道	札幌市中央区	北三条西4丁目1番地第一生命ビル
6	1101	60	608570	{null}	{null}	{null}	北海道	札幌市中央区	南一条西14丁目
7	1101	60	608630	{null}	{null}	{null}	北海道	札幌市中央区	北二条東17丁目2番地
8	1101	60	608540	{null}	{null}	{null}	北海道	札幌市中央区	北一条東9丁目99番地19
9	1101	60	608613	{null}	{null}	{null}	北海道	札幌市中央区	北一条西4丁目2-2札幌ノースプラザ1
10	1101	60	608542	{null}	{null}	{null}	北海道	札幌市中央区	北二条東4丁目1-2
11	1101	60	608537	{null}	{null}	{null}	北海道	札幌市中央区	南一条西4丁目20番地
12	1101	60	608619	{null}	{null}	{null}	北海道	札幌市中央区	北四条西2丁目1
13	1101	60	608509	{null}	{null}	{null}	北海道	札幌市中央区	北三条西1丁目10番地東芝札幌ビル6F
14	1101	60	608552	{null}	{null}	{null}	北海道	札幌市中央区	北一条西6丁目2番地
15	1101	60	608508	{null}	{null}	{null}	北海道	札幌市中央区	南二条西2丁目13番地
16	1101	60	608521	{null}	{null}	{null}	北海道	札幌市中央区	北三条東3丁目1-30
17	1101	60	608797	{null}	{null}	{null}	北海道	札幌市中央区	北二条西4丁目3番地
18	1101	60	608637	{null}	{null}	{null}	北海道	札幌市中央区	南二条西5丁目メゾン本府701号
19	1101	60	608605	{null}	{null}	{null}	北海道	札幌市中央区	北一条西6丁目1番地2アーバンネット札
20	1101	60	608532	{null}	{null}	{null}	北海道	札幌市中央区	北一条西2丁目1番地札幌時計台ビル
21	1101	60	608797	{null}	{null}	{null}	北海道	札幌市中央区	北二条西4丁目3番地
22	1101	60	608568	{null}	{null}	{null}	北海道	札幌市中央区	南二条西4丁目6
23	1101	60	608623	{null}	{null}	{null}	北海道	札幌市中央区	南一条西1丁目1番地」CB札幌東ビル
24	1101	60	608577	{null}	{null}	{null}	北海道	札幌市中央区	北一条西6丁目1-2アーバンネット札幌
25	1101	60	608555	{null}	{null}	{null}	北海道	札幌市中央区	北四条西3丁目1
26	1101	60	608626	{null}	{null}	{null}	北海道	札幌市中央区	北一条西3丁目札幌メダルビル2 F

SQL を用いた更新 (Update using SQL)

SQL を用いた**データの更新 (update)**の実行例を示す.「**UPDATE <table-name> SET <attribute-name>=<expression> WHERE <expression>**」の形をした SQL は, データの更新である. この SQL 文を「BEGIN TRANSACTION」と「COMMIT」で囲む. (″UPDATE ... SET ...″ means database update).

1. SQL プログラムの記述

「UPDATE ... SET ...」は更新. ここには 1**つの SQL 文**を書き,「BEGIN TRANSACTION」と「COMMIT」で囲む. (″UPDATE ... SET ...[″] means database update.

```
BEGIN TRANSACTION:
UPDATE E SET score=90, updated_at=datetime('now')
WHERE name = 'Database' AND student_name ='KK';
COMMIT;
```

🥌 mydb - Sqliteman		
<u>File Context Database</u>	<u>S</u> ystem	Help
Sche <u>m</u> a <u>P</u> ragmas		
Database main Tabl··· - Im Tabl··· - bundles - e	1 2 3 4	BEGIN TRANSACTION; UPDATE E SET score=90, updated_at=datetime(now') WHERE name = 'Database' AND student_name ='KK'; 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</u> <u>C</u> ontext <u>D</u> atabase	<u>S</u> ystem	Help
Sche <u>m</u> a <u>P</u> ragmas		R 💌 📼 🗁 🏝 🗟
Database in im main in Tabl··· in bundles in e	1 2 3 4	BEGIN TRANSACTION; UPDATE E SET score=90, updated_at=datetime(now') WHERE name = 'Database' AND student_name ='KK'; 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

1 Script started 2 ■ BEGIN TRANSACTION; 3 No error 4 5 UPDATE E SET score=90, updated_at=datetime(now') 6 WHERE name = 'Database' AND student_name ='KK'; 7 No error 8 9 COMMIT; 10 No error 11	Full View I Item View Script Output	
12 Script finished	1 Script started 2 BEGIN TRANSACTION; 3 No error 4 5 UPDATE E SET score=90, updated_at=datetim 6 WHERE name = 'Database' AND student_name 7 No error 8 9 COMMIT; 10 No error 11 12 Script finished	ne(now') e ='KK:

4. E テーブル のブラウズ (Browse table 'E')

Data Data	abase abase	90	КК	2009-12-14 01:24:40	2009-12-14 01:31:55
2 Data	ahase				
		95	AA	2009-12-14 01:24:40	{null}
3 Data	abase	80	LL	2009-12-14 01:24:40	{null}
1 Prog	gramming	85	кк	2009-12-14 01:24:40	{null}
5 Prog	gramming	75	LL	2009-12-14 01:24:40	{null}

5. SQL プログラムの記述

「UPDATE ... SET ...」は更新. ここには 1つの SQL 文を書き,「BEGIN TRANSACTION」と「COMMIT」で囲む. ("UPDATE ... SET ..." means database update.

```
BEGIN TRANSACTION:
UPDATE E SET score=80, updated_at=datetime('now')
WHERE name = 'Programming' AND student_name = 'LL';
COMMIT:
  mydb - Sqliteman
<u>File C</u>ontext <u>D</u>atabase <u>S</u>ystem <u>H</u>elp
                                       2 🕪
                                                                                   Q
                                                   Sche<u>m</u>a <u>P</u>ragmas
    Database
                              1
                                     BEGIN TRANSACTION
                                       UPDATE E SET score=80, updated_at=datetime('now')
WHERE name = 'Programming' AND student_name ='LL';
                               2
3
4
    🖻 🗐 main
        🔆 💷 Tabl•••
```

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

⊕- bundles

COMMIT:

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)

🥌 mydb - Sqliteman		
<u>File Context</u> <u>D</u> atabase	<u>S</u> ystem	Help
Sche <u>m</u> a <u>P</u> ragmas		🖹 🕪 🖻 🖺 🗁 🏝 🔍
Database im g main im Tabl··· im Tabl···	1 2 3 4	BEGIN TRANSACTION; UPDATE E SET score=80, updated_at=datetime(now') WHERE name = 'Programming' AND student_name ='LL'; COMMIT;

- 7. 「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 UPDATE E SET score=80, updated_at=datetim 6 ₩HERE name = 'Programming' AND student_rid 7 No error 8 9 COMMIT; 10 No error 11 12 Script finished	e(now) ame ='LL':

8. **E テーブル** のブラウズ (Browse table 'E')

	Full View I Item View Script Output						
		name	score	student_name	created_at	updated_at	
	1	Database	90	кк	2009-12-14 01:24:40	2009-12-14 01:31:55	
	2	Database	95	AA	2009-12-14 01:24:40	{null}	
	3	Database	80	LL	2009-12-14 01:24:40	{null}	
	4	Programming	85	КК	2009-12-14 01:24:40	{null}	
(5	Programming	80	LL	2009-12-14 01:24:40	2009-12-14 01:38:26	
						1	

SQL を用いた行の削除 (Delete row(s) using SQL)

SQL を用いた行の削除 (delete row(s))の実行例を示す.

「**DELETE FROM <table-name> WHERE <expression>;**」の形をした SQL は行の削除である. この SQL 文を「BEGIN TRANSACTION」と「COMMIT」で囲む. ("DELETE FROM ... WHERE ..." means deletion of rows).

1. SQL プログラムの記述

```
BEGIN TRANSACTION
DELETE FROM E
WHERE name = 'Database' AND student_name = 'AA' :
COMMIT:
 Smydb - Sqliteman
<u>File Context</u> <u>Database</u> <u>System</u> <u>H</u>elp
                                                📑 🗁 🏝 🔍
                          🕨 🦹 🕪 📑
  Sche<u>m</u>a <u>P</u>ragmas
                               BEGIN TRANSACTION
   Database
                         2
3
4
                                DELETE FROM E
   Ė 📒 main
                                WHERE name = 'Database' AND student_name ='AA';
      🔄 📰 Tabl…
                                COMMIT
          🕀 bundles
```

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 View I Item View Script Output					

4. **E テーブル** のブラウズ (Browse table 'E')

Fu	Full View Item View Script Output						
Г	name	score	student_name	created_at	updated_at		
1	Database	90	КК	2009-12-14 01:24:40	2009-12-14 01:31:55		
2	Database	80	LL	2009-12-14 01:24:40	{null}		
3	Programming	85	кк	2009-12-14 01:24:40	{null}		
4	Programming	80	LL	2009-12-14 01:24:40	2009-12-14 01:38:26		
F							

ジャーナルファイルの確認 (Examine a journal file)

1. SQL プログラムの記述

今度は COMMIT を書かない (Without COMMIT)

BEGIN TRANSACTION; INSERT INTO E VALUES('Database', 90, 'BB', datetime('now'), NULL); INSERT INTO E VALUES('Programming', 95, 'BB', datetime('now'), NULL);	
🛇 mydb - Sqliteman	
<u>File Context Database System H</u> elp	
Sche <u>ma</u> Pragmas 🕨 🖹 III 🔤 🗋 🖆 🏝 🔍	
Database 1 E BEGIN TRANSACTION; INSERT INTO E VALUES('Database', 90, 'BB', datetime('now'), NULL); INSERT INTO E VALUES('Programming', 95, 'BB', datetime('now'), NULL); INSERT INTO E VALUES('Programming', 95, 'BB', datetime('now'), NULL);	

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</u> <u>Database</u> <u>System</u>	Help
Sche <u>m</u> a <u>P</u> ragmas	
Database 1	FIBEGIN TRANSACTION
i⊡- ⊜ main i⊡- Tabl··· i⊡- bundles	INSERT INTO E VALUES ('Database', 90, 'BB', datetime (now'), NULL); INSERT INTO E VALUES ('Programming', 95, 'BB', datetime (now'), NULL);

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 E VALUES('Database', 90, ' 6 No error 7 8 INSERT INTO E VALUES('Programming', 95, 9 9 No error 10 11 Script finished 12	BB', datetime(now'), NULL); 'BB', datetime(now'), NULL);

4. ジャーナル・ファイルの確認 (Journal File)

C:¥SQLiteを開く、「mydb-journal」がジャーナル・ファイルである.

C:#SQLite			
ファイル(E) 編集(E) 表示(⊻)	お気に入り(<u>A</u>) ツール(T) ヘルプ(H)		
🔆 F3 • 🕥 • 🏂 🔎	検索 🦻 フォルダ 🛛 🏠 🏂 🗙	9	
名前 🔺	サイズ 種類	更新日時	
🖬 mydb	17 KB ファイル	2009/12/14 10:59	
salite3.exe	513 KB アプリケーション	2009/12/07 12:05	
🗔 mydb-journal	4 KB ファイル	2009/12/14 11:00)
			-

5. Sqliteman で COMMIT の発行 (COMMIT)

COMMIT

S mydb - Sqliteman	
<u>File Context</u> <u>D</u> atabase	System <u>H</u> elp
Sche <u>m</u> a <u>P</u> ragmas	
Database	

- 6. 「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



7. ジャーナル・ファイルの確認 (Journal File)

C:¥SQLite を開く. COMMIT を発行した. データベースの修正は終わっている. そのためジャーナルファイルが消えている.

ĺ	🔁 C:¥SQLite			
	ファイル(E) 編集(E) 表示(⊻)	お気に入り(<u>A</u>) ツール(T) ヘルプ(H)		
	🔆 F3 • 🕥 • 🏂 🔎	検索 🌔 フォルダ 🛛 📴 🗙	19	
ſ	名前 🔺	サイズ 種類	更新日時	
ſ	📷 mydb	17 KB ファイル	2009/12/14 11:08	
	ra sqlite3.exe	513 KB アプリケーション	2009/12/07 12:05	

ROLLBACK の確認 (Example a journal file)

1. SQL プログラムの記述

今度も COMMIT を書かない (Without COMMIT)

BEGIN TRANSACTION: INSERT INTO E VALUES('Database', 95, 'CC', datetime('now') INSERT INTO E VALUES('Programming', 80, 'CC', datetime('now')	, NULL): , NULL):
S mydb - Sqliteman	
<u>File Context</u> <u>D</u> atabase <u>S</u> ystem <u>H</u> elp	
Sche <u>m</u> a Pragmas	
Database Database 1 BEGIN TRANSACTION; 2 INSERT INTO E VALUES('Database', 3 INSERT INTO E VALUES('Programming)	95, CC', datetime('now'), NULL); ng', 80, 'CC', datetime('now'), NULL);

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)

Sn	nydb - Sqlit	teman		
<u>F</u> ile	<u>C</u> ontext	<u>D</u> atabase	<u>S</u> ystem	Help
S	iche <u>m</u> a	<u>P</u> ragmas		🖹 🕞 🔄 🔄 🔁 🕲
-	Database È-Berna È-Berna	in Tabl•••	1 2 3	BEGIN TRANSACTION INSERT INTO E VALUES('Database', 95, (CC', datetime('now'), NULL); INSERT INTO E VALUES('Programming', 80, 'CC', datetime('now'), NULL);

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

1 Script started 2 ■ BEGIN TRANSACTION; 3 No error 4 5 INSERT INTO E VALUES('Database', 95, 'CC', datetime('now'), NULL); 6 No error 7 8 INSERT INTO E VALUES('Programming', 80, 'CC', datetime('now'), NULL); 9 No error 10 11 Script finished 12	

4. ジャーナル・ファイルの確認 (Journal File disappears)

C.¥SQLite を開く、「mydb-journal」がジャーナル・ファイルである.

🗁 C:¥SQLite			
ファイル(E) 編集(E) 表示(⊻)	お気に入り(<u>A</u>) ツール(T) へ)	レプ(圧)	
📀 戻る • 🕥 • 🏂 🔎	検索 🍺 フォルダ 🛛 🏠 🗿	> 🗙 🍤 🛄	
名前 🔺	サイズ 種類	更新日時	
📷 mydb	17 KB ファイル	2009/12/14 11:08	
🕞 mydb-journal	4 KB ファイル	2009/12/14 11:13	
📩 sqlite3.exe	513 KB アプリケーショ	ン 2009/12/07 12:05	

5. E テーブル のブラウズ (Browse table 'E')

「INSERT INTO **E** VALUES('Database', 95, 'CC', datetime('now'), NULL);」,「INSERT INTO **E** VALUES ('Programming', 80, 'CC', datetime('now'), NULL);」の結果がデータベースに反映されているように見える. (It seems that database update is finished)

Full View Item View Script Output						
$\left[\right]$	name	score	student_name	created_at	updated_at	
1	Database	90	кк	2009-12-14 01:24:40	2009-12-14 01:31:55	
2	Database	80	LL	2009-12-14 01:24:40	{null}	
3	Programming	85	кк	2009-12-14 01:24:40	{null}	
4	Programming	80	LL	2009-12-14 01:24:40	2009-12-14 01:38:26	
5	Database	90	BB	2009-12-14 02:00:50	{null}	
6	Programming	95	BB	2009-12-14 02:00:50	{null}	
7	Database	95	CC	2009-12-14 02:13:43	{null}	
8	Programming	80	CC	2009-12-14 02:13:43	{null}	

6. Sqliteman で **ROLLBACK の発行** (ROLLBACK)

今度は ROLLBACK を発行する. (Issue ROLLBACK)

ROLLBACK

S mydb - Sqliteman	
<u>File Context</u> Database	System <u>H</u> elp
Sche <u>m</u> a <u>P</u> ragmas	🜔 😰 🕪 📼 🗀 🗁 🏝 🔍
Database 🖻 🚔 main	1 ROLLBACK

7. 「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



8. データベース・ファイルとジャーナル・ファイルの確認 (Journal File disappears)

実はデータベースファイルは一切修正されていない (Database file is unchaned. It is observerd by using the file timestamp).

ļ				
	ファイル(E) 編集(E) 表示(V)	お気に入り(A) ツール(T) ヘル:	Ĵ(<u>H</u>)	
	🕞 戻る • 🕥 • 🏂 🔎	検索 🍺 フォルダ 🎼 🎲 爹	× 🍤 💷	
	名前 🔺	サイズ 種類	更新日時	
	mydb	17 KB ファイル	2009/12/14 11:08	
	sqlite3.exe	513 KB アプリケーション	2009/12/07 12:05	

9. E テーブル のブラウズ (Browse table 'E')

F	ull View	Item View	Script Out	tput		
ſГ		name	score	student_name	created_at	updated_at
1	Databas	se i	90	кк	2009-12-14 01:24:40	2009-12-14 01:31:55
2	Databas	e	80	LL	2009-12-14 01:24:40	{null}
3	Program	nming	85	КК	2009-12-14 01:24:40	{null}
4	Program	nming	80	LL	2009-12-14 01:24:40	2009-12-14 01:38:26
Ε	Databas	e -	90	BB	2009-12-14 02:00:50	{null}
Ē	Program	nming	95	BB	2009-12-14 02:00:50	{null}
¥						·

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)

次の PTABLE テーブルに関する問題 (About the following 'PTABLE' table)

name	type	color
apple	fruit	red
apple	fruit	blue
rose	flower	white
rose	flower	red
rose	flower	yellow

1. PTABLE を次のように更新する SQL を書きなさい (Write a SQL which updates the table).

name	type	color
app e app e	fruit fruit	red blue
rose	flower	white
rose	flower	red
rose	flower	blue

2. さらに, PTABLE から行を削除し, 次のテーブルに変更する SQL を書きなさい (Write a SQL which delete a row).

name	type	color
apple	fruit	red
rose	flower	white
rose	flower	red
rose	flower	blue

3. SQLite を使い、下記の SQL を評価させなさい (Evaluate the following SQL)

```
(1)
CREATE TABLE GG (
id INTEGER PRIMARY KEY,
name TEXT):
(2)
BEGIN TRANSACTION:
INSERT INTO GG values (1, 'X'):
COMMIT:
select * from GG:
(3)
BEGIN TRANSACTION:
INSERT INTO GG values (2, 'Y'):
ROLLBACK:
select * from GG:
```

解答例 (Answers)

1. UPDATE PTABLE SET color='blue' WHERE name='rose' AND type='flower' AND color='yellow':

2. DELETE FROM PTABLE WHERE name='apple' AND type='fruit' AND color='blue';