آزمایشگاه پایگاه داده

رحمت قاسمی

۲۴ آبان ۱۳۹۶

فهرست مطالب

فصل ۱

مقدمه

۱.۱ تعاریف اولیه

۱.۱.۱ پایگاه داده

تعاریف متعددی برای پایگاه داده در مراجع مختلف وجود دارد با این حال یک تعریف ساده از آن به این صورت است که پایگاه داده[۱] یک مجموعه از داده‌های ذخیره شده در ارتباط با یک سازمان می‌باشد. پایگاه داده توسط نرم‌افزار مدیریت پایگاه داده‌ها (DBMS[۲]) مدیریت می‌شود و از طریق آن در اختیار سایر نرم‌افزارها و یا کاربران قرار می‌گیرد. بنابراین می‌توان تصور نمود هرگونه تغییر در داده‌ها و یا ساختار پایگاه داده‌ها از طریق نرم افزار مدیریت پایگاه داده انجام می‌شود. بانک اطلاعاتی معادل دیگری برای پایگاه داده است.

۲.۱.۱ دامنه

یک مجموعه از مقادیر ممکن برای یک صفت خاص، دامنه نامیده می‌شود. مثلا ممکن است صفتی به نام ماه تولد داشته باشیم. ماه تولد می‌تواند هر یک از مقادیر به صورت مجموعه زیر باشد.

$$D_{month} = \{۱, ۲, ۳, ۴, ۵, ۶, ۷, ۸, ۹, ۱۰, ۱۱, ۱۲\}$$

۳.۱.۱ رابطه

رابطه زیرمجموعه‌ای از ضرب دکارتی دو یا چند دامنه است. یک رابطه را می‌توان به صورت یک جدول نمایش داد. به صورتی که هر ردیف در جدول یک عضو از رابطه را که به آن تاپل می‌گوییم نشان می‌دهد و هر یک از ستونهای جدول یک صفت از رابطه را مشخص می‌کند. بنابراین می‌توانیم رابطه را با جدول معادل بنامیم.

در پایگاه داده‌ای رابطه‌ای جدول اساسی‌ترین شی قابل استفاده برای ذخیره اطلاعات در پایگاه داده است. این بدان معنی است که حداقل در یک پایگاه داده رابطه‌ای یکسری جداول وجود دارد

---

[۱] Data Base

[۲] Data Base Management System

که اطلاعات سازمان در آنها نگهداری می‌شود.

۲.۱ کلیدها

۱.۲.۱ فوق کلید
۲.۲.۱ کلید کاندید
۳.۲.۱ کلید اصلی
۴.۲.۱ کلید بدیل
۵.۲.۱ کلید خارجی

۳.۱ معرفی چند DBMS

نرم‌افزارهای مختلفی برای مدیریت پایگاه داده‌ها وجود دارد از آن جمله می‌توان به نرم افزارهای زیر اشاره نمود.

- MySql
- Sql Server
- Oracle
- Informix
- Interbase
- Db2
- Ingress

هر چند نرم‌افزار SQlite را نمی‌توان به صورت قطعی به عنوان یک DBMS محسوب کرد ولی با توجه به سادگی آن محیط مناسب برای یادگیری مفاهیم اولیه پایگاه داده‌ها می‌باشد.

فصل ۲

دستور کار ۱

۱.۲ نصب برنامه

بسته نرم افزاری مورد نیاز برای انجام آزمایشها SQLite می‌باشد. در لوح فشرده همراه نرم‌افزار و فایلهای مورد نیاز برای آزمایشات که در ادامه مطرح خواهد شد، وجود دارد. همچنین نرم افزار SQLite بسادگی از طریق وب سایت https://www.sqlite.org/download.html قابل دانلود است و در سیستم عاملهای مختلف نرم‌افزار مربوط به آن یافت می‌شود.

۲.۲ اجرای SQLite

بعد از اتمام موفقیت آمیز مراحل قبلی اکنون شما قادر هستید برنامه SQLite را اجرا نمایید. برای اجرا ابتدا وارد محیط command شده و سپس دستور زیر را در جلوی خط فرمان تایپ کنید:

```
1 c:\> sqlite3 test.db
```

که test.db نام فایل مربوط به پایگاه داده شما می‌باشد و می‌توانید از نام دیگری استفاده نمایید. بعد از اجرای موفق SQLite نتیجه‌ای مشابه زیر را شاهد خواهید بود.

```
1 SQLite version 3.8.2 2013-12-06 14:53:30
2 Enter ".help" for instructions
3 Enter SQL statements terminated with a ";"
4 sqlite>
```

۳.۲ اجرای دستورات SQL در SQLite

برای اجرای دستورات SQL کافی است دستور مورد نظر را در خط فرمان SQLite وارد کنید. دقت نمایید که در پایان دستور باید علامت ؛ (سمی‌کالن) قرار دهید و بعد از آن کلید ENTER را بزنید تا دستور شما اجرا و نتیجه نمایش داده شود. برای مثال دستور SELECT 1+2*3 را وارد کنید تا نتیجه زیر را مشاهده نمایید.

```
sqlite> SELECT 1+2*3;
7
sqlite>
```

۴.۲ تمرین

تمرینات زیر را به ترتیب انجام دهید تا نتیجه مطلوب حاصل شود. نتیجه هر یک از تمرینات را در گزارش کار خود یادداشت نمایید.

تمرین ۲.۱. *محتویات لوح فشرده را در درایو C کپی نمایید.*

■

تمرین ۲.۲. *برنامه sqlite3.exe را اجرا نمایید.*
شما اکنون وارد محیط دستوری SQLite شده‌اید و قادر خواهید دستورات مورد نظر را وارد کنید.

■

تمرین ۲.۳. *دستورات زیر به ترتیب وارد نمایید و نتیجه اجرای هر دستور را یادداشت نمایید. این دستورات برخی از دستورات خود SQLite است. دستورات SQLite با علامت . آغاز می‌شوند.*

```
.databases
.open test.db
.databases
.tables
.help
```

■

تمرین ۲.۴. *دستور زیر وارد نموده و نتیجه اجرا را یادداشت نمایید.*

```
SELECT 1+2*3;
```

■

فصل ۳
دستور کار ۲

۱.۳ پایگاه داده فرضی

در این آزمایشگاه از پایگاه داده فرضی ساده مربوط به یک فروشگاه استفاده شده است. این پایگاه داده دارای یکسری جداول است که در جدول ۱.۳ لیست شده است. نام این پایگاه داده را test.db در نظر می‌گیریم. هر یک از جداول این پایگاه داده در بخش ۳.۳ تعریف شده است.

SQL یک زبان ساخت‌یافته پرس‌وجو می‌باشد. این بدان معنی است که SQL رابطی بین برنامه و پایگاه داده است. پایگاه داده SQLite از SQL پشتیبانی می‌کند. در ادامه برخی از دستورات SQL که در نرم‌افزار SQLite قابل استفاده است به طور خلاصه معرفی می‌شود. در صورت نیاز به شکل کامل دستورات به راهنمای SQLite مراجعه نمایید.

۲.۳ ایجاد پایگاه داده

SQLite از طریق خط فرمان نام پایگاه داده را درخواست می‌کند و در صورتی که نباشد آن را ایجاد می‌نماید.

```
1  c:\> sqlite3 test.db
```

در مثال فوق اگر فایل test.db یافت نشود SQLite آن را ایجاد کرده و برای استفاده در نظر می‌گیرد.

جدول ۱.۳: پایگاه داده فرضی با نام forosh

نام جدول	شرح	فیلدها
customer	مشتری	کدمشتری، نام مشتری، شهر، آدرس، تلفن
good	کالا	کد کالا، نام کالا، قیمت واحد، موجودی انبار
order	سفارش	کدسفارش، کدمشتری، تاریخ سفارش، وضعیت سفارش
orderDetail	قلم سفارش	کدسفارش، کدکالا، تعداد، قیمت واحد

جدول ۲.۳: برخی از انواع داده‌ها در SQLite

نوع داده	شرح
INT	مقدار عددی صحیح
BIGINT	مقدار عددی صحیح بزرگ
FLOAT	مقدار عددی اعشاری
CHAR(n)	رشته کاراکتری به طول n
VARCHAR(n)	رشته کاراکتری با طول متغییر حداکثر n
DATE	تاریخ
TIME	ساعت
DATETIME	تاریخ و ساعت

۳.۳ ایجاد جدول

دستور CREATE TABLE یک جدول جدید در پایگاه داده انتخاب شده ایجاد می‌کند. این دستور به صورت زیر قابل استفاده است.

```
CREATE TABLE tablename (
    field1 datatype1 [DEFAULT value] [NOT NULL],
    field2 datatype2 [DEFAULT value] [NOT NULL],
    ...
    PRIMARY KEY (field(s))
);
```

در مدل دستوری فوق tablename نام جدول جدید است که ممکن است شامل تعدادی فیلد باشد که با field مشخص می‌شوند. برای هر فیلد یک نوع داده یا datatype باید مشخص شود. برخی از انواع داده‌های قابل استفاده در جدول ۲.۳ مشخص شده است. به کمک DEFAULT مقدار پیش‌فرض فیلد را مشخص می‌کنیم و NOT NULL مشخص می‌کند که فیلد مربوط به آن نمی‌تواند هیچ‌مقدار پذیر باشد و حتما باید در هنگام ورود اطلاعات مقدار آن مشخص باشد. PRIMARY KEY برای مشخص کردن کلید اصلی جدول استفاده می‌شود. کلید اصلی ممکن است شامل تعدادی فیلد باشد که با کاراکتر کاما از یکدیگر جدا می‌شوند.

مثال ۳.۱. *ایجاد جدول مشتری*

```
1 CREATE TABLE customer(
2         customerId INT NOT NULL,
3         customerName VARCHAR(30),
4         city     VARCHAR(20),
5         address  VARCHAR(60),
6         telNo    VARCHAR(12),
7         PRIMARY KEY (customerId)
8 );
```

■

مثال ۳.۲. *ایجاد جدول کالا*

```
CREATE TABLE good(
        goodId INT NOT NULL,
        goodName VARCHAR(30),
        unitPrice INT,
        stockValue INT,
        PRIMARY KEY (goodId)
);
```

■

مثال ۳.۳. *ایجاد جدول سفارش*

```
CREATE TABLE orders(
        orderId INT NOT NULL,
        customerId INT NOT NULL,
        orderDate DATETIME,
        orderState CHAR(10),
        PRIMARY KEY (orderId)
);
```

■

مثال ۳.۴. *ایجاد جدول قلم سفارش*

```
CREATE TABLE orderDetail(
        orderId INT NOT NULL,
        goodId INT NOT NULL,
        qty INT,
        unitPrice INT,
        PRIMARY KEY (orderId,goodId)
);
```

■

۴.۳ تمرین

تمرینات زیر را به ترتیب انجام دهید تا نتیجه مطلوب حاصل شود. نتیجه هر یک از تمرینات را در گزارش کار خود یادداشت نمایید.

تمرین ۳.۱. *برنامه sqlite را اجرا کنید. در صورت لزوم به تمرین فصل قبلی مراجعه کرده و تا تمرین ۲.۲ را انجام دهید.* ■

تمرین ۳.۲. *دستور زیر را وارد کنید. این دستور لیست جداول موجود در دیتابیس را نشان می‌دهد. خروجی این دستور را یادداشت کنید.*

```
1 .tables
```

■

تمرین ۳.۳. *یک جدول به نام کتاب در دیتابیس ایجاد نمایید. برای این کار دستور زیر را وارد کنید. جدول کتاب حاوی مشخصات کتاب مانند کدکتاب، نام کتاب، نام مولف و قیمت کتاب است.*

```
1 CREATE TABLE book(
2         book_id INT NOT NULL,
3         book_name VARCHAR(100),
4         book_author VARCHAR(50),
5         book_price  INT,
6         PRIMARY KEY (book_id)
7 );
```

■

تمرین ۳.۴. *دستور زیر را وارد کنید. اگر همه چیز درست باشد، شما بایستی نام جدولی که در تمرین قبلی ایجاد کرده‌اید را مشاهده نمایید.*

```
1 .tables
```

■

فصل ۴

دستور کار ۳

۱.۴ تغییر ساختار جدول

دستور ALTER TABLE برای تغییر ساختار جدول استفاده می‌شود. در ساده ترین حالت این تغییر ممکن است حذف یک فیلد، اضاف کردن یک فیلد و یا تغییر یک فیلد یاشد. تغییرات دیگری نیز وجود دارد که در اینجا بدان اشاره نشده است. شکل دستور ALTER TABLE به صورت زیر است:

```
1 ALTER TABLE tablename ADD  newfield definition;
2 ALTER TABLE tablename DROP  col_name;
3 ALTER TABLE tablename CHANGE oldfield_name newfield_name definition;
```

ممکن است در اجرای دستور ردیف دوم و سوم در sqlite با مشکل روبرو شوید. چون این دستورات در این زمان پشتیبانی نمی‌شود.

۲.۴ درج اطلاعات در جدول

درج اطلاعات در جداول پایگاه داده توسط دستور INSERT انجام میشود. شکل کلی این دستور به صورت زیر است.

```
1 INSERT INTO tablename(field1, field2, ...)
2                 VALUES (value1, value2, ...);
```

در دستور INSERT، tablename نام جدولی است که عمل درج در آن انجام خواهد شد. بعد از VALUES لیست فیلدهایی از جدول را که قرار است مقدار آنها را در هنگام درج مشخص نماییم می‌نویسم، لذا به ترتیب فبلدها مقادیر متناسب با آنها را بعد از VALUES لیست می‌کنیم. اگر لیست فیلدها مشخص نشود، لیست مقادیر متناسب با لیست فیلدهای جدول در هنگام ایجاد آن، به ترتیب در لیست مقادیر نوشته میشود.

مثال ۴.۱. درج در *جدول کالا*

```
1 INSERT INTO good(goodId, goodName, unitPrice)
2                   VALUES( 1, 'Hard␣Disk', '197000');
```

■

مثال ۴.۲. درج چند کالا در جدول کالا

```
1 INSERT INTO good(goodId, goodName, unitPrice)
2         VALUES( 2, 'CPU␣FAN', '16000'),
3         ( 3, 'RAM␣256MB', '28000'),
4         ( 4, 'External␣Modem', '56000');
```

■

مثال ۴.۳. درج در جدول مشتری

```
1 INSERT INTO Customer
2         VALUES( 1, 'Reza', 'Neka,␣Enghelab␣Ave.', '09110001234');
```

در این مثال با توجه به اینکه بعد از نام جدولی فیلدی مشخص نشده است بنابراین لیست مقادیر با توجه به فیلدهای جدول *Customer* در هنگام ایجاد ارایه شده است. ■

مثال ۴.۴. درج سفارش جدید با شماره ۵ در جدول سفارش برای مشتری شماره۱

```
1 INSERT INTO order( orderId, customerId, orderDate)
2         VALUES(5, 1, NOW() );
```

در مثال فوق سفارش شماره ۵ برای مشتری شماره ۱ ثبت می‌شود. تابع *NOW()* زمان جاری سیستم را نشان می‌دهد. ■

۳.۴ تمرین

تمرین ۴.۱. برنامه *sqlite3* را اجرا نمایید. ■

تمرین ۴.۲. به کمک دستور *open*. پایگاه داده فرضی *test.db* را باز کنید. ■

تمرین ۴.۳. به کمک دستور *tables*. جداول پایگاه داده فرضی را بازبینی نمایید. اگر جداول مربوطه یافت نشد احتمالا دیتابیس را اشتباه انتخاب نمودید. ■

تمرین ۴.۴. دستورات زیر را به ترتیب وارد کنید.

```
.mode column
.headers on
```

■

تمرین ۴.۵. دستور زیر را وارد کنید تا محتویات جدول *customer* را مشاهده نمایید.

```
SELECT * FROM customer;
```

■

تمرین ۴.۶. یک ستون جدید به نام *status* اضافه کنید، برای این کار دستور زیر را وارد کنید.

```
ALTER TABLE customer ADD COLUMN status INT;
```

به کمک دستور *SELECT* در تمرین قبلی نتیجه کار خود را مشاهده نمایید. ■

تمرین ۴.۷. دستور *schema.* را وارد کنید. این دستور ساختار پایگاه داده را به شما نشان می‌دهد. اگر بخواهید ساختار یک جدول را مشاهده کنید کافی است نام جدول را بعد از دستور *schema.* بنویسید.

```
.schema customer
```

■

تمرین ۴.۸. به کمک دستور *INSERT* ردیفهای را در جدول مشتری درج کنید. برای راهنمایی بیشتر به مثالهای این بخش مراجعه نمایید. ■

تمرین ۴.۹. ستون *status* از جدول مشتری را حذف کنید. ■

فصل ۵

دستور کار ۴

۱.۵ تغییر و بروزرسانی اطلاعات

برای تغییر اطلاعات جدول از دستور UPDATE استفاده می‌شود. شکل کلی این دستور به صورت زیر است.

```
1 UPDATE tablename
2         SET field1=value1,
3         fiel2=value2,
4         ...
5         [WHERE condition];
```

مثال ۵.۱. *تغییر قیمت کالای شماره ۵*

```
1 UPDATE good
2         SET unitPrice=200
3         WHERE goodId=5;
```

■

مثال ۵.۲. *ده درصد کاهش قیمت برای همه کالاها*

```
1 UPDATE good
2         SET unitPrice=unitPrice-0.1*unitPrice;
```

■

۲.۵ حذف اطلاعات

دستور DELETE، ردیف یا ردیفهایی را از جدول حذف می‌کند. شکل کلی این دستور به صورت زیر است.

```
1 DELETE FROM tablename
2         [WHERE condition];
```

ردیفهایی از جدول tablename حذف می‌شوند که با شرط conditon مطابقت داشته باشند. اگر شرط مشخص نشود، تمام ردیفهای جدول حذف می‌شوند.

مثال ۵.۳. *حذف کالا با شماره ۵ از سفارش شماره ۹*

```
1 DELETE FROM orderDetail
2         WHERE orderId=9 AND goodId=5;
```

■

۳.۵ حذف جدول

دستور DROP TABLE برای حذف جدول از پایگاه داده استفاده می‌شود. شکل کلی این دستو به صورت زیر است.

```
1 DROP TABLE tablename;
```

tablename نام جدولی است که قرار است حذف شود. اطلاعات و ساختار جدول به طور کامل از پایگاه داده حذف می‌شود. در استفاده از این دستور باید مراقب بود که جدول به اشتباه حذف نشود.

۴.۵ تمرین

تمرین ۵.۱. *برنامه sqlite3 را اجرا نمایید.* ■

تمرین ۵.۲. *به کمک دستور open. پایگاه داده فرضی test.db را باز کنید.* ■

تمرین ۵.۳. *دستورات زیر را اجرا کنید.*

```
1 .headers on
2 .mode column
```

■

بعد از انجام هریک از تمرینهای زیر نتیجه کار خود را به کمک دستور SELECT بررسی نمایید.

تمرین ۵.۴. *افزایش ده درصدی قیمت به تمام کالاها را انجام دهید.* ■

تمرین ۵.۵. *حذف کالای شماره ۵ از سفارش شماره ۹.* ■

تمرین ۵.۶. *حذف تمام کالاها از سفارش شماره ۸.* ■

تمرین ۵.۷. *در سفارش شماره ۹ از کالای شماره ۵ به تعداد ۲۸ خریداری شده است.* ■

تمرین ۵.۸. *فرض کنید که در سفارش شماره ۹ قبلا از کالای شماره ۵ به تعداد ۲۸ خریداری شده است و اکنون ۵ عدد نیز از همان کالا به این سفارش اضافه شده است. جداول قلم سفارش را بروزرسانی کنید.* ■

تمرین ۵.۹. *وضعیت سفارش شماره ۹ نهایی میشود. با این فرض که با نهایی شدن سفارش، به ازای هر کالا از موجودی انبار به تعداد کالای انتخاب شده، کم می‌شود.* ■

فصل ۶
دستور کار ۵

۱.۶ دستور SELECT

برای بازیابی اطلاعات از پایگاه داده از دستور SELECT استفاده می‌شود. یادگیری این دستور از اهمیت بالایی برخوردار است. شکل کلی این دستور به صورت زیر است. قابل ذکر است که برای مشاهده مدل کاملتر این دستور به راهنمای SQLite مراجعه نمایید.

```
1 SELECT [DISTINCT] expr1, expr2, ...
2         [FROM table_references
3         [WHERE condition]
4         [GROUP BY col1, col2, ...
5         [HAVING (condition)]]
6         [ORDER BY col_name [ASC | DESC],...]]
```

در ادامه هر یک از بخشهای مطرح شده در شکل کلی دستور SELECT معرفی می‌شود.

- DISTINCT، ردیفهای تکراری در جدول خروجی را حذف می‌کند.
- FROM table_references، ارجاعی به جداول در پایگاه داده است. در حالت ساده نام تعدادی از جداول است که با کاراکتر کاما (،) جدا شده‌اند.
- WHERE condition، شرط انتخاب ردیفها را مشخص می‌کند. condition یک عبارت شرطی را مشخص می‌کند.
- GROUP BY برای گروهبندی و تفکیک اطلاعات خروجی استفاده می‌شود.
- HAVING(condition)، شرط روی گروهها را مشخص می‌کند. این بخش به هنگام استفاده از GROUP BY ممکن است استفاده شود. condition یک عبارت شرطی را مشخص می‌کند.
- ORDER BY، برای مرتب سازی خروجی استفاده می‌شود. مرتب‌سازی ممکن است بر اساس یک یا چند فیلد یا عبارت باشد. ASCبرای مرتب سازی صعودی و DESC برای مرتب سازی نزولی است.

برای درک بیشتر مثالهای متعددی از دستور SELECT در اینجا ارایه خواهد شد.

۲.۶ پرسشهای ساده

مثال ۶.۱. *لیست نام و شماره تلفن مشتریان*

```
1 SELECT customerName, telNo
2         FROM customer
```

■

در دستور SELECT امکان محاسبه عبارات محاسباتی هم وجود دارد. به مثال زیر توجه کنید.

مثال ۶.۲. جمع دو عدد

```
1 SELECT 1+2;
```

■

مثال ۶.۳. *لیست مشتریان ساکن نکا*

```
1 SELECT customerId, customerName, city, Address, telNo
2         FROM customer
3         WHERE city='neka'
```

■

مثال ۶.۴. *لیست شهرهای مربوط به مشتریان*

```
1 SELECT DISTINCT city
2         FROM customer
```

■ *برای حذف مقادیر تکراری از DISTINCT استفاده شده است.*

اگر همه ستونهای جدول مد نظر باشد به جای نوشتن نام تمام فیلدها می‌توان از کاراکتر * استفاده نمود.

مثال ۶.۵. *لیست مشتریان ساکن نکا*

```
1 SELECT *
2         FROM customer
3         WHERE city='neka'
```

■

۳.۶ تمرین

تمرین ۶.۱. *برنامه sqlite3 را اجرا نمایید.* ■

تمرین ۶.۲. *دستورات زیر را اجرا کنید.*

```
.open test.db
.headers on
.mode column
```

■

برای هریک از تمرینهای زیر یک دستور SELECT مناسب بنویسید.

تمرین ۶.۳. *لیست کالاهایی که موجودی انبار آنها از ۲۰ کمتر است.*

■

تمرین ۶.۴. *لیست مشتریان ساکن Mashhad.*

■

تمرین ۶.۵. *کد سفارشهای مربوط به مشتری با شماره ۲.*

■

تمرین ۶.۶. *کدکالاهایی که در سفارش شماره ۵ وجود دارد.*

■

تمرین ۶.۷. *لیست کالاهایی که قیمت آنها از ۱۰۰ بیشتر و از ۲۰۰ کمتر است.*

■

تمرین ۶.۸. *کد مشتریانی که حداقل یک سفارش ثبت شده دارند (با حذف تکرار).*

■

تمرین ۶.۹. *کد کالاهایی که حداقل یک بار سفارش داده شده‌اند (با حذف تکرار).*

■

فصل ۷
دستور کار ۶

۱.۷ گروهبندی داده‌ها

با استفاده از GROUP BY می‌توان خروجی را گروهبندی نمود. در هنگام گروهبندی استفاده از توابع جمعی ممکن است مد نظر باشد. توابع جمعی عبارتند از:

- SUM، برای محاسبه مجموع استفاده می‌شود.
- COUNT، شمارش یک مجموعه از داده‌ها.
- MIN، کمترین مقدار را محاسبه می‌کند.
- MAX، بیشترین مقدار را محاسبه می‌کند.

مثال ۷.۱. *تعداد مشتریان به تفکیک شهر*

```
1 SELECT city, count(*)
2         FROM customer
3         GROUP BY city;
```

■

مثال ۷.۲. *جمع کل مبلغ سفارشها به تفکیک*

```
1 SELECT orderId, SUM(qty*unitPrice)
2         FROM orderDetail
3         group by orderId
```

■

۲.۷ استفاده از HAVING

در استفاده از GROUP BY می‌توان از HAVING برای اعمال شرط روی گروهها استفاده نمود. به مثال زیر توجه نمایید.

مثال ۷.۳. *کدکالاهایی که بیشتر از یک مورد در لیست سفارشات ثبت شده‌اند*

```
1 SELECT gooId
2         FROM orderDetail
3         group by goodId
4         HAVING(count(*) > 1)
```

■

۳.۷ مرتب سازی خروجی

برای مرتب کردن نتیجه می‌توان از ORDER BY در دستور SELECT استفاده نمود. بعد از ORDER BY نام فیلد یا فیلدهایی که می‌خواهیم مرتب سازی بر اساس آن انجام شود را می‌نویسیم. نوع مرتب سازی را می‌توان به کمک ASC و یا DESC مشخص نمود. ASC مرتب سازی صعودی و DESC، مرتب سازی نزولی را مشخص می‌کند.

مثال ۷.۴. *لیست نام و شماره تلفن مشتریان به ترتیب نام مشتری*

```
1 SELECT customerName, telNo
2         FROM customer
3         ORDER BY customerName ASC
```

■

مرتب سازی خروجی ممکن است بر اساس چند فیلد باشد.

مثال ۷.۵. *لیست مشتریان به ترتیب شهر و نام*

```
1 SELECT *
2         FROM customer
3         ORDER BY city, customerName
```

■

۴.۷ تمرین

تمرین ۷.۱. *برنامه sqlite3 را اجرا نمایید.* ■

تمرین ۷.۲. *دستورات زیر را اجرا کنید.*

```
1    .open test.db
2    .headers on
```

```
3 .mode column
```

برای هریک از تمرینهای زیر یک دستور SELECT مناسب بنویسید.

تمرین ۷.۳. *مجموع تعداد کالاهای سفارش داده شده به تفکیک.*

تمرین ۷.۴. *مجموع کالاهای سفارش داده شده در سفارش شماره ۵.*

تمرین ۷.۵. *تعداد سفارشهاس داده شده توسط هر مشتری به تفکیک.*

تمرین ۷.۶. *جمع مبلغ مربوط به سفارش شماره ۵.*

تمرین ۷.۷. *تعداد سفارشها به تفکیک و ترتیب شماره مشتری.*

تمرین ۷.۸. *کدمشتریانی که بیشتر از یک مورد سفارش ثبت شده دارند.*

فصل ۸

دستور کار ۷

۱.۸ عملگرها

در SQLite عملگرهای متعددی روی داده‌ها تعریف شده است. عملگرهای تعریف شده در جدول ۳.۸ لیست شده است.

در این مثال از عملگر LIKE برای تست شباهت رشته‌ای استفاده می‌شود. در استفاده از این عملگر دو کاراکتر جادویی را می‌توان استفاده نمود.

- % این کاراکتر نشان دهنده چند حرف از رشته است.
- _ این کاراکتر نشان دهنده یک حرف ناشناخته از رشته است.

مثال ۸.۱. *لیست مشتریان که نام آنها با حرف B شروع می‌شود.*

```
1 SELECT *
2         FROM customer
3         WHERE customerName LIKE 'B%';
```

■

مثال ۸.۲. *لیست مشتریانی که حرف دوم نام آنها A است. چون حرف اول نامشخص است از کاراکتر _ برای اولین حرف استفاده می‌کنیم.*

جدول ۱.۸: عملگرهای محاسباتی درSQLite

عملگر	شرح
+	جمع
-	تفریق
*	ضرب
/	تقسیم
%	باقیمانده

جدول ۲.۸: عملگرهای مقایسه‌ای درSQLite

عملگر	شرح
=, ==	عملگر تساوی
>=	بزرگتراز یا مساوی با
>	بزرگتر از
<=	کوچکتر یا مساوی
<	کوچکتر از
!=, <>	مخالف بودن

جدول ۳.۸: عملگرهای منطقیSQLite

عملگر	شرح
AND	عملگر AND برای ترکیب شرایط مختلف در دستور WHERE استفاده می‌شود.
BETWEEN	تست وجود یک مقدار در یک محدوده
EXISTS	عملگر EXISTS مشخص کننده وجود سطر در جدول یا عبارت داده شده است.
IN	عملگر IN یک مقدار را در مجموعه‌ای از مقادیر جستجو می‌کند.
NOT IN	برخلاف عملگر IN عمل می‌کند.
LIKE	عملگر LIKE برای بررسی شباهت رشته‌ای استفاده می‌شود. این عملگر به حروف کوچک و بزرگ حساس نیست.
GLOB	عملگر GLOB برای بررسی شباهت رشته‌ای استفاده می‌شود. این عملگر به حروف کوچک و بزرگ حساس است.
NOT	عملگر NOT به همراه سایر عملگرهای منطقی استفاده می‌شود و عملکرد آنها را نقیض می‌کند.
OR	عملگر OR برای ترکیب شرایط مختلف در دستور WHERE استفاده می‌شود.
IS NULL	عملگر NULL برای مقایسه یک مقدار با مقدار NULL کاربرد دارد.
IS	طرز کار عملگر IS مشابه عملگر تساوی است.
IS NOT	عملگر IS NOT، همانند هملگر نابرابری عمل می‌کند.
||	دو عبارت رشته‌ای را به یکدیگر ملحق می‌کند.
UNIQUE	عملگر UNIQUE هر یک از ردیفهای جدول داده شده را به منظور تکراری نبودن جستجو می‌کند.

جدول ۴.۸: عملگرهای بیتی درSQLite

عملگر	شرح
&	و بیتی
|	یای بیتی
~	نقیض بیتی
<<	شیفت به چپ
>>	شیفت به راست

```
1 SELECT *
2         FROM customer
3         WHERE customerName LIKE '_A%';
```

■

مثال ۸.۳. *لیست کالاهایی که قیمت آنها از ۱۰۰ تا ۱۵۰ است.*

```
1 SELECT *
2         FROM good
3         WHERE unitPrice BETWEEN 100 AND 150;
```

■

مثال ۸.۴. *لیست مشتریانی که آدرس آنها نامشخص است*

```
1 SELECT *
2         FROM customer
3         WHERE address IS NULL;
```

■

۲.۸ تمرین

تمرین ۸.۱. *برنامه sqlite3 را اجرا نمایید.*

■

تمرین ۸.۲. *دستورات زیر را اجرا کنید.*

```
1    .open test.db
2    .headers on
3    .mode column
```

■

تمرین ۸.۳. *نتیجه هریک از عبارتهای زیر را بنویسید.*

```
SELECT 1+2;
SELECT 10 % 3;
SELECT 1 IN (2,3,4,5);
SELECT 1 IN (2,3,4,5,1);
SELECT 'ali' LIKE 'al*';
SELECT 'ali' GLOB 'AL*';
SELECT 4 << 1;
SELECT 4 << 2;
SELECT 4 << 3;
SELECT 25 << 1;
SELECT 50 >> 1;
SELECT 50 >> 2;
SELECT 125 BETWEEN 100 AND 200;
SELECT 99 BETWEEN 100 AND 200;
SELECT 100 BETWEEN 100 AND 200;
SELECT 200 BETWEEN 100 AND 200;
```

■

برای هریک از تمرینهای زیر یک دستور SELECT مناسب بنویسید.

تمرین ۸.۴. *لیست مشتریانی که حداقل یک سفارش ثبت کرده‌اند. به کمک عملگر EXISTS.* ■

تمرین ۸.۵. *لیست مشتریانی که حداقل یک سفارش ثبت کرده‌اند. به کمک عملگر IN.* ■

تمرین ۸.۶. *لیست مشتریانی که هیچ سفارش ثبت شده‌ای ندارند.* ■

فصل ۹
دستور کار ۸

۱.۹ پیوند جداول

در بیشتر پرس‌وجوها لازم است خروجی مورد نیاز از پیوند دو یا چند جدول استخراج شود. در ساده‌ترین حالت نام جداول بعد از FROM نوشته می‌شود و شرط پیوند در قسمت WHERE مشخص می‌شود.

مثال ۹.۱. *فاکتور مربوط به صورتحساب شماره ۵ به همراه نام کالا*

```
1 SELECT o.goodId, g.goodName, o.qty, g.unitPrice,
2    o.qty*g.unitPrice AS price
3    FROM orderDetail o, good g
4    WHERE orderId=5 AND (o.goodId=g.goodId);
```

■ *در این مثال از AS برای نامگذاری ستون محاسبه شده استفاده شده است.*

مثال ۹.۲. *تعداد کالاهایی که مشتری با شماره ۸ خریداری کرده است.*

```
1 SELECT COUNT(*)
2         FROM order o, orderDetail od
3         WHERE o.customerId=8 AND (o.orderId=od.orderId);
```

■ *در این مثال برای ساده شدن بکارگیری نام جداول از نام مستعار برای هر جدول استفاده شده است. o برای جدول order و od برای جدول orderDetail.*

۲.۹ تابع IN

به کمک تابع IN می‌توان وجود یک مقدار در یک مجموعه را بررسی نمود.

مثال ۹.۳. *لیست کالاهایی که قیمت آنها از ۱۰۰ یا ۱۲۰ است*

```
1 SELECT *
2         FROM good
3         WHERE unitPrice IN(100,120);
```

■

۳.۹ تمرین

تمرین ۹.۱. *برنامه sqlite3 را اجرا نمایید.*

تمرین ۹.۲. *دستورات زیر را اجرا کنید.*

```
1  .open test.db
2  .headers on
3  .mode column
```

برای هریک از تمرینهای زیر یک دستور SELECT مناسب بنویسید.

تمرین ۹.۳. *لیست مشتریانی که کالای شماره ۲ را خریداره کرده‌اند.*

تمرین ۹.۴. *لیست کالاهایی که در یک تاریخ خاص مثلا (۲۰۱۵-۰۸-۱۲) خریداری شده‌اند. شامل (کدکالا، نام کالا، تعدادخرید)*

فصل ۱۰

دستور کار ۱۳

۱.۱۰ ارتباط نرم افزار با دیتابیس

در این بخش به نحوه ارتباط نرم‌افزار به پایگاه داده اشاره می‌شود. برای این منظور زبان C و زبان PHP انتخاب شده‌اند.

۲.۱۰ ارتباط با پایگاه داده Sqlite از زبان C

برای ارتباط با پایگاه داده Sqlite و استفاده از آن در زبان C سرفایل sqlite.h وجود دارد. در ادامه یک برنامه به زبان C برای این منظور نوشته شده است. برخی از توابع مربوط به SQlite که در این برنامه استفاده شده است در جدول ؟؟ معرفی شده‌اند.

```
1  #include <stdio.h>
2  #include <string.h>
3  #include <sqlite3.h>
4
5  static int callback(void *data, int argc, char **argv, char **attr) {
6     int i;
7     static int e=0;
8     for(i=0; i<argc; i++)
9        printf("%s\t",  argv[i] ? argv[i] : "<NULL>");
10    printf("\n");
11    return 0;
12 }
13
14 int main(int argc, char* argv[]) {
15    sqlite3 *db;
16    char *error = 0;
17    int rc;
18    char sql[1000];
```

```
19
20    rc = sqlite3_open("test.db", &db);
21
22    if( rc ){
23       fprintf(stderr, "Can't␣open␣database:␣%s\n", sqlite3_errmsg(db));
24       return(0);
25    }else{
26       fprintf(stderr, "Opened␣database␣successfully\n");
27    }
28
29    printf("Enter␣Sql␣Command:␣");
30    fgets(sql,1000,stdin);
31
32    rc = sqlite3_exec(db, sql, callback, 0 , &error);
33    if( rc != SQLITE_OK ){
34       fprintf(stderr, "SQL␣error:␣%s\n", error);
35       sqlite3_free(error);
36    }else{
37       fprintf(stdout, "Operation␣done␣successfully\n");
38    }
39    sqlite3_close(db);
40    return 0;
41 }
```