C5a:

Introduction to SQLite

SQL (Structured Query Language)

- A computer language aim to store, manipulate, and retrieve data in (relational) databases

- The de facto standard in traditional database applications and DBMS

Structured Query Language

- **Data Definition Language (DDL)**
  - *Creating, altering and deleting tables and other database objects*
- **Data Manipulation Language (DML)**
  - *Inserting, updating and deleting rows*
  - *Querying tables*
- **Database Control language (DCL)**
  - *Controlling access to the database*
- **Transaction Control Language (TCL)**
  - *Dealing with transactions within a database.*

Common SQL statements

- SELECT (DML)
- CREATE (DDL)
- INSERT (DML)
- UPDATE (DML)
- DELETE (DML)
- DROP (DDL)

CREATE A DATABASE

To create a database to collect all the tables.

```
CREATE DATABASE
      school.db;
```

Sample Statement

```
CREATE DATABASE
      database_name;
```

(Simplified) Syntax

CREATE A TABLE

To create a table ‘`student`’ with six columns

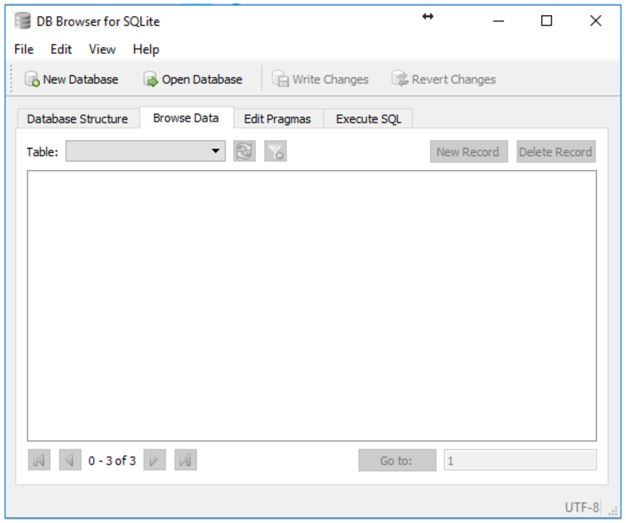
Field Name	Data Type
name	VARCHAR(32)
studentNo	CHAR(8)
year	INTEGER
matricDate	DATE
faculty	VARCHAR(64)
department	VARCHAR(32)

Common Data Types

- VARCHAR[*n*]
  - Variable-length string having maximum *n* characters.
- CHAR[*n*]
  - Fixed-length string of *n* characters.
- INTEGER
- DATE
  - *“YYYY-MM-DD”*

Demo : A DBMS with a sample database

- DB Browser for SQLite
  - A visual tool for creating, designing and editing SQLite databases
  - http://sqlitebrowser.org/



CREATE A TABLE

To create a table ‘`student`’ with six columns

```
CREATE TABLE student (
   name VARCHAR(32),
   studentNo CHAR(8),
   year INTEGER,
   matricDate DATE,
   faculty VARCHAR(64),
   department
VARCHAR(32)
);
```

Sample Statement

```
CREATE TABLE table_name (
   column_name1 type1,
   column_name2 type2,
   …
   column_nameN typeN
);
```

(Simplified) Syntax

INSERT DATA

To insert one new student into the table

Field Name	Data
name	`'Jie Jie'`
studentNo	`'A123456X'`
year	`3`
matricDate	`'2015-01-01'`
faculty	`'School of Computing'`
department	`'Computer Science'`

INSERT

To insert one new student into the table

```
INSERT INTO student
VALUES (
  'Jie Jie',
  'A123456X',
  3,
  '2015-01-01',
  'School of Computing',
  'Computer Science');
```

Sample Statement

```
INSERT INTO table_name
VALUES (
   value1,
   value2,
   …
   valueN
);
```

(Simplified) Syntax

INSERT

What happen if some of the values are missing / unknown?

- Use a NULL value instead.

```
INSERT INTO student
VALUES (
  'Jie Jie',
  'A123456X',
  3,
  '2015-01-01',
  'School of Computing',
 NULL);
```

INSERT DATA

To insert two more new students into the table

Field Name	Data
name	`'Dewi Wijaj'`
studentNo	`'B234567Y'`
year	`4`
matricDate	`'2014-01-01'`
faculty	`'School of Computing'`
department	`'Computer Engineering'`

INSERT DATA

To insert two more new students into the table

Field Name	Data
name	`'Xie Xin'`
studentNo	`'C345678Z'`
year	`3`
matricDate	`'2015-04-01'`
faculty	`'Faculty of Science'`
department	`'Chemistry'`

INSERT

To insert multiple new students into the table

Sample Statement

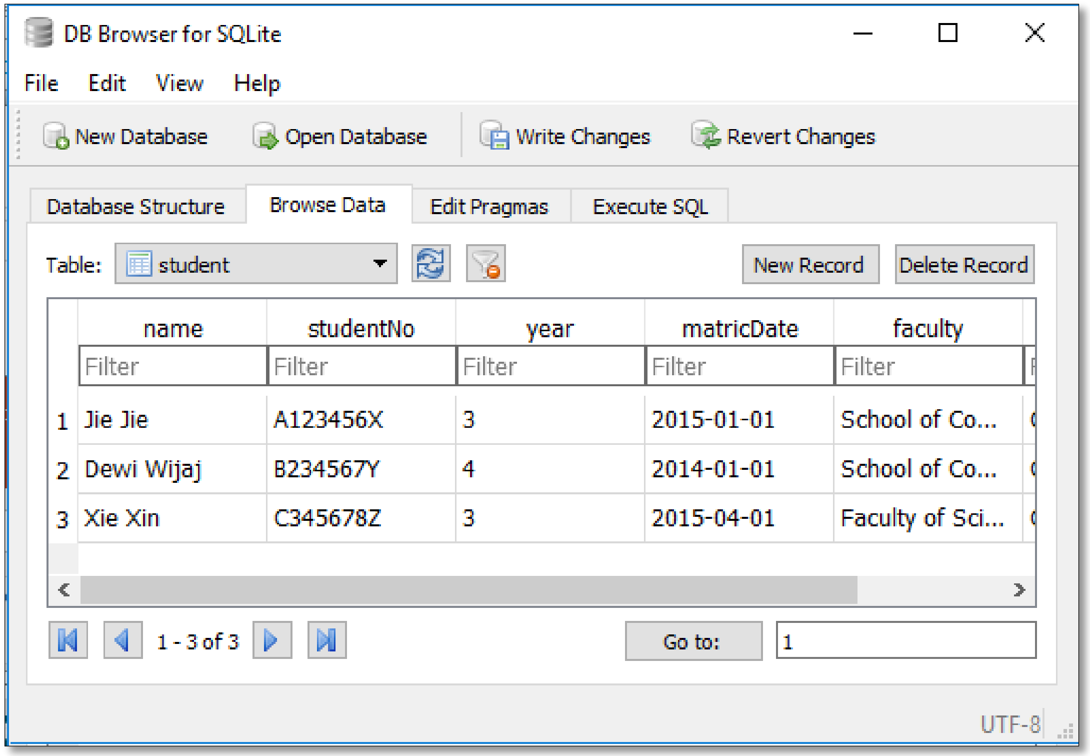
```
INSERT INTO student
VALUES
('Dewi Wijaj','B234567Y',4,'2014-01-01',
'School of Computing','Computer Engineering'),
('Xie Xin','C345678Z',3,'2015-04-01',
'Faculty of Science','Chemistry');
```

(Simplified) Syntax

```
INSERT INTO table_name
VALUES
(Data1a, Data1b, Data1c ... Data1z),
(Data2a, Data2b, Data2c ... Data2z)
... (Data9a, Data9b, Data9c ... Data9z);
```

Demo : A DBMS with a sample database

- "Browse Data" tab

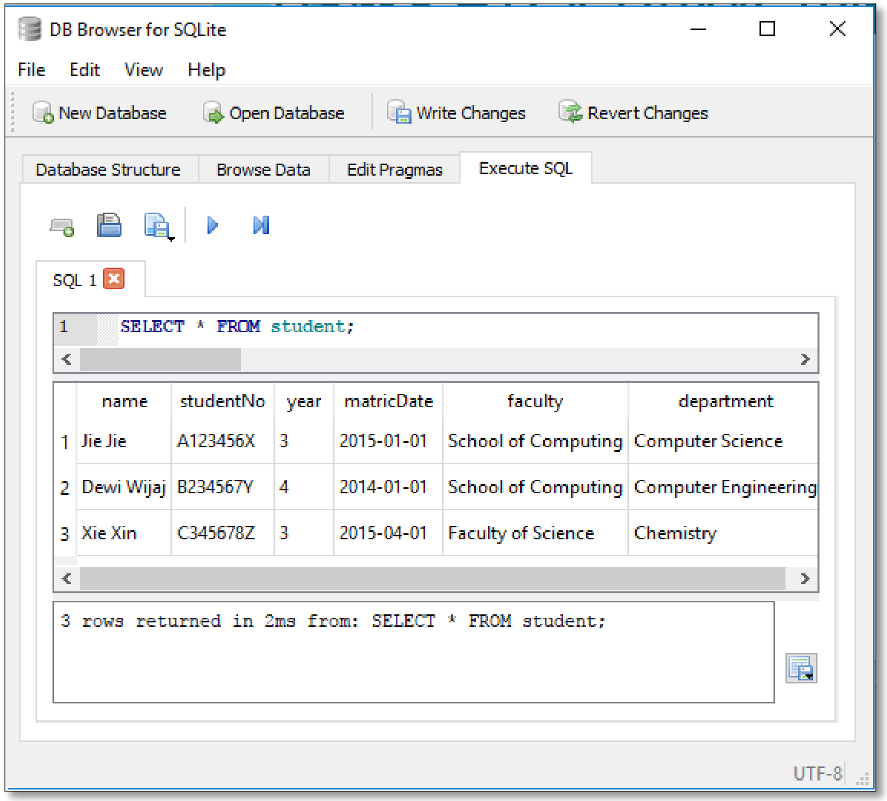


	name	studentNo	year	matricDate	faculty
1	Jie Jie	A123456X	3	2015-01-01	School of Co...
2	Dewi Wijaj	B234567Y	4	2014-01-01	School of Co...
3	Xie Xin	C345678Z	3	2015-04-01	Faculty of Sci...

school.db

Demo : A DBMS with a sample database

- “Execute SQL” tab



DB Browser for SQLite
File Edit View Help
New Database Open Database Write Changes Revert Changes
Database Structure Browse Data Edit Pragmas Execute SQL
SQL 1
UTF-8

```
SELECT * FROM student;
```

	name	studentNo	year	matricDate	faculty	department
1	Jie Jie	A123456X	3	2015-01-01	School of Computing	Computer Science
2	Dewi Wijaj	B234567Y	4	2014-01-01	School of Computing	Computer Engineering
3	Xie Xin	C345678Z	3	2015-04-01	Faculty of Science	Chemistry

```
3 rows returned in 2ms from: SELECT * FROM student;
```

```
SELECT *
FROM student;
```

SELECT

To return all data in a table

```
SELECT * FROM student;
```

Sample Statement

```
SELECT * FROM table_name;
```

(Simplified) Syntax

To return selected columns in a table

```
SELECT name, year
FROM student;
```

Sample Statement

```
SELECT col_name1,col_name2,
…, col_nameN
FROM table_name;
```

(Simplified) Syntax

SELECT

To return certain rows in a table

```
SELECT * FROM student
WHERE department =
'Computer Science';
```

Sample Statement

```
SELECT * FROM table_name
WHERE condition;
```

(Simplified) Syntax

UPDATE

To update the values in a table for all rows.

```
UPDATE student
SET year = 1;
```

Sample Statement

```
UPDATE student
SET assignment;
```

(Simplified) Syntax

The update can make use of the values before the update to define the values after the update.

Sample Statement

```
UPDATE student
SET year = year + 1;
```

UPDATE

To update the values in a table selectively

```
UPDATE student
SET department = 'CS'
WHERE department =
  'Computer Science';
```

Sample Statement

```
UPDATE student
SET assignment;
WHERE condition;
```

(Simplified) Syntax

DELETE

To remove all data in a table.

```
DELETE FROM student;
```

To remove selected data in a table.

```
DELETE FROM student
  WHERE department =
   'Computer Science';
```

Sample Statement

```
DELETE FROM table_name
  WHERE condition;
```

(Simplified) Syntax

DROP

To remove a table (and all the data in it).

```
DROP TABLE student;
```

- What is wrong with this table?

name	studentNo	year	matricDate
Jie Jie	A123456Y	3	3500-01-01
Dewi Wijaj	A123456Y	4	2014-01-01
NULL	A222333Z	-1	2015-04-01

Integrity Constraints

- A mechanism to ensure accuracy and consistency of the data in a relational database.
- Common integrity constraints
  - NOT NULL
  - UNIQUE
  - CHECK
  - PRIMARY KEY
  - FOREIGN KEY
- These constraints, if specified, are automatically checked by the DBMS

NOT NULL

The value of an attribute cannot be missing / unknown (i.e., the use of the NULL value is forbidden.)

- The quantity of the item purchased must be present in an order

NOT NULL

- Single-attribute

```
CREATE TABLE book (
  title VARCHAR(256),
  authors VARCHAR(256),
  publisher VARCHAR(64),
  ISBN13 CHAR(14) NOT NULL);
```

UNIQUE

The combination of the values of one or more attributes must be unique.

- The mobile phone numbers of different people must be unique.

UNIQUE

- Single-attribute

```
CREATE TABLE book (
  title VARCHAR(256),
  authors VARCHAR(256),
  publisher VARCHAR(64),
  ISBN13 CHAR(14) UNIQUE);
```

- Multi-attribute

```
CREATE TABLE book (
  title VARCHAR(256),
  authors VARCHAR(256),
  publisher VARCHAR(64),
  ISBN13 CHAR(14),
  UNIQUE(title, authors));
```

CHECK

Any requirements that must be met by the attribute values.

- The format of a book must be "paperback" or "hardcover"
- The start time of an event must be earlier than its end time

CHECK

```
CREATE TABLE book (
  title VARCHAR(256),
  authors VARCHAR(256),
  publisher VARCHAR(64),
  ISBN13 CHAR(14),
  CHECK (length(ISBN13)=14));
```

CHECK

```
CREATE TABLE products (
    product_id   INTEGER NOT NULL,
    product_name TEXT NOT NULL,
    list_price   DECIMAL (10, 2) NOT NULL,
    discount     DECIMAL (10, 2) DEFAULT 0,
    CHECK (list_price >= discount AND
        discount >= 0 AND
        list_price >= 0)
);
```

PRIMARY KEY

A set of attributes that identifies uniquely a tuple:

- People - national identification number, email address, first name and last name
- Flights - Airline name and flight number

- PRIMARY KEY = NOT NULL + UNIQUE

PRIMARY KEY

- What does it imply if "title" is said to be the primary key of this table?
- Is "title" a good primary key?
- What other columns may serve as a primary key?

title	authors	publisher	ISBN13
The Future of Learning Institutions in a Digital Age	Cathy N. Davidson, David Theo Goldberg	The MIT Press	978-0262513593
Introduction to Algorithms	Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein	The MIT Press	978-0262033848
The Digital Photography Book	Scott Kelby	Peachpit Press	978-0321474049
Computer Organization and Design	David A. Patterson, John L. Hennessy	Morgan Kaufmann	978-0123744937

PRIMARY KEY

- Single-attribute

```
CREATE TABLE book (
  title VARCHAR(256),
  authors VARCHAR(256),
  publisher VARCHAR(64),
  ISBN13 CHAR(14) PRIMARY KEY);
```

```
CREATE TABLE book (
  title VARCHAR(256),
  authors VARCHAR(256),
  publisher VARCHAR(64),
  ISBN13 CHAR(14),
  PRIMARY KEY (ISBN13));
```

PRIMARY KEY

- Multi-attribute

```
CREATE TABLE book (
  title VARCHAR(256),
  authors VARCHAR(256),
  publisher VARCHAR(64),
  ISBN13 CHAR(14),
  PRIMARY KEY (title, authors));
```

FOREIGN KEY

- The values of this set of attributes must refer to an existing tuple in another relation.
  - The ISBN of a book in a loan record must belong to an existing book in book relation.

Loan

name	ISBN13
Jie Jie	978-0262513593
Xie Xin	978-0262033848
Jie Jie	978-0123744937

Book

title	authors	publisher	ISBN13
The Future of Learning Institutions in a Digital Age	Cathy N. Davidson, David Theo Goldberg	The MIT Press	978-0262513593
Introduction to Algorithms	Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein	The MIT Press	978-0262033848
The Digital Photography Book	Scott Kelby	Peachpit Press	978-0321474049
Computer Organization and Design	David A. Patterson, John L. Hennessy	Morgan Kaufmann	978-0123744937

FOREIGN KEY

- Based on the following book relation,

```
CREATE TABLE book (
   title VARCHAR(256),
   authors VARCHAR(256),
   publisher VARCHAR(64),
   ISBN13 CHAR(14) PRIMARY KEY);
```

we can create the loan relation as follows:

```
CREATE TABLE loan (
   name VARCHAR(256),
   ISBN13 CHAR(14) REFERENCES book(ISBN13));
```

FOREIGN KEY

Alternatively, if we have the following book relation,

```
CREATE TABLE book (
   title VARCHAR(256),
   authors VARCHAR(256),
   publisher VARCHAR(64),
   ISBN13 CHAR(14),
   PRIMARY KEY (title, authors));
```

we can create the loan relation as follows:

```
CREATE TABLE loan (
   student VARCHAR(256),
   title VARCHAR(256),
   authors VARCHAR(256),
   FOREIGN KEY (title, authors) REFERENCES
book (title, authors));
```

FOREIGN KEY

- Cascading can be implemented for such violations
  - If a book is updated / deleted, the related loan records are also updated / deleted.

Loan

name	ISBN13
Jie Jie	978-0262513593
Xie Xin	978-0262033848
Jie Jie	978-0123744937

Book

title	authors	publisher	ISBN13
The Future of Learning Institutions in a Digital Age	Cathy N. Davidson, David Theo Goldberg	The MIT Press	978-0262513593
Introduction to Algorithms	Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein	The MIT Press	978-0262033848
The Digital Photography Book	Scott Kelby	Peachpit Press	978-0321474049
Computer Organization and Design	David A. Patterson, John L. Hennessy	Morgan Kaufmann	978-0123744937

Multi-table SELECT Statements

- Find the name of the student who borrowed "Introduction to Algorithms".

Loan

name	ISBN13
Jie Jie	978-0262513593
Xie Xin	978-0262033848
Jie Jie	978-0123744937

Book

title	authors	publisher	ISBN13
The Future of Learning Institutions in a Digital Age	Cathy N. Davidson, David Theo Goldberg	The MIT Press	978-0262513593
Introduction to Algorithms	Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein	The MIT Press	978-0262033848
The Digital Photography Book	Scott Kelby	Peachpit Press	978-0321474049
Computer Organization and Design	David A. Patterson, John L. Hennessy	Morgan Kaufmann	978-0123744937

```
SELECT loan.name FROM book, loan
WHERE book.ISBN13 = loan.ISBN13 AND
book.title = 'Introduction to Algorithms';
```

Multi-table SELECT Statements

- Table names can be renamed to simplify the query.

```
SELECT loan.name FROM book, loan
WHERE book.ISBN13 = loan.ISBN13 AND
book.title = 'Introduction to Algorithms';
```

```
SELECT loan.name FROM book b, loan l
WHERE b.ISBN13 = l.ISBN13 AND
b.title = 'Introduction to Algorithms';
```

Multi-table SELECT Statements

- Table names can be omitted if there is no ambiguity.

```
SELECT loan.name FROM book, loan
WHERE book.ISBN13 = loan.ISBN13 AND
book.title = 'Introduction to Algorithms';
```

```
SELECT name FROM book, loan
WHERE book.ISBN13 = loan.ISBN13 AND
title = 'Introduction to Algorithms';
```