

```
CREATE TABLE Employee(  
ID          VARCHAR(4) NOT NULL,  
First_Name  VARCHAR(10),  
Last_Name   VARCHAR(10),  
Start_Date  DATE,  
End_Date    DATE,  
Salary      Decimal(8,2),  
City        VARCHAR(10),  
Description  VARCHAR(15)  
); // 顧員(ID,名字,姓氏,開始,結束日期 yyyy-mm-dd,薪金,城市,描述)
```

ID	First_Name	Last_Name	Start_Date	End_Date	Salary	City	Description
01	Jason	Martin	2010-01-27	2011-04-27	1234.56	Toronto	Programmer
02	Alison	Mathews	2010-01-27	2011-04-27	6661.78	Vancouver	Tester
03	James	Smith	2010-01-27	2011-04-27	6544.78	Vancouver	Tester
04	Celia	Rice	2010-01-27	2011-04-27	2344.78	Vancouver	Manager
05	Robert	Black	2010-01-27	2011-04-27	2334.78	Vancouver	Tester
06	Linda	Green	2010-01-27	2011-04-27	4322.78	New York	Tester
07	David	Larry	2010-01-27	2011-04-27	7897.78	New York	Manager
08	James	Cat	2010-01-27	2011-04-27	1232.78	Vancouver	Tester

```
// 插入紀錄
```

```
insert into Employee  
  values ('01','Jason','Martin','19960725','20060725',1234.56,'Toronto','Programmer');  
insert into Employee  
  values('02','Alison','Mathews','19760321','19860221',6661.78,'Vancouver','Tester');  
insert into Employee  
  values('03','James','Smith','19781212','19900315',6544.78,'Vancouver','Tester');  
insert into Employee  
  values('04','Celia','Rice','19821024','19990421',2344.78,'Vancouver','Manager');  
insert into Employee  
  values('05','Robert','Black','19840115','19980808',2334.78,'Vancouver','Tester');  
insert into Employee  
  values('06','Linda','Green','19870730','19960104',4322.78,'New York','Tester');  
insert into Employee  
  values('07','David','Larry','19901231','19980212',7897.78,'New York','Manager');  
insert into Employee  
  values('08','James','Cat','19960917','20020415',1232.78,'Vancouver','Tester');
```

```
SELECT salary*2 "Double salary" FROM employee;
```

// strncpy = substr 子字串

```
SELECT first_name, SUBSTR(first_name,1,2) MGR_LOCATION,  
       SUBSTR(first_name,2,2) ITEM_NUMBER  
FROM employee;
```

create table temp as

```
       select sum(salary) as salarytotal from employee // total salary 總薪金  
SELECT id, first_name, salary, salary/salarytotal // ?%  
       FROM employee, temp // join 2 tables  
drop table temp
```

SELECT * FROM employee WHERE id > ANY

(select id from employee where id in ('02', '03', '04')); // 03,04,05,... (大於其中一個 id)

SELECT * FROM employee WHERE id > ALL

(select id from employee where id in ('02', '03', '04')); // 05,06,07,... (大於所有 id)

select 'is true ' where 1=1 or 1=0 and 0=1; // true

select 'is true ' where (1=1 or 1=0) and 0=1; // false

select 'is true ' where 1=1 or (1=0 and 0=1); // true

SELECT 2*6;

SELECT 10*(12/3-1) AS "Computation";

SELECT first_name FROM employee WHERE salary >= 30000 AND start_date IS NOT NULL ORDER BY first_name;	SELECT first_name, salary*3+1 FROM employee; SELECT first_name, salary+2 FROM employee;
---	--

SELECT * FROM employee WHERE id BETWEEN '01' AND '03';

SELECT * FROM employee WHERE first_name BETWEEN 'A' AND 'K';

SELECT * FROM employee

WHERE start_date BETWEEN '19980415' and '20050212';

SELECT id, salary, <u>if(salary>4500,'HIGH','LOW')</u> AS status FROM employee; // id, salary, status('HIGH','LOW')	SELECT id, salary, <u>CASE</u> <u>WHEN salary>4500 THEN 'HIGH'</u> <u>ELSE 'LOW'</u> <u>END AS status</u> FROM employee;
---	--

```
SELECT city, count(*)
FROM employee
GROUP BY city
```

<pre>SELECT count(*), <u>CASE</u> city WHEN 'Vancouver' THEN 'No' WHEN 'New York' THEN 'Yes' WHEN null THEN 'Null' <u>END</u> city FROM employee GROUP BY city; // 5 No, 2 Yes, 1 Null</pre>	<pre>SELECT id, <u>CASE</u> id WHEN '01' THEN 'A' WHEN '02' THEN 'B' WHEN '03' THEN 'C' WHEN '04' THEN 'D' <u>ELSE</u> 'Default' <u>END</u> as grade FROM employee;</pre>
---	--

```
SELECT concat( first_name, ' ', last_name ) AS "Customer Name"
FROM employee
```

<pre>SELECT Description FROM employee SELECT <u>DISTINCT</u> Description FROM employee SELECT city, count(*) FROM employee; // <u>error</u> // 與 count(*) 結果一樣 SELECT city, count(first_name) FROM employee <u>GROUP BY city</u>; // 按 city 分組 SELECT <u>count</u>(*) FROM employee <u>GROUP BY city</u>; // 各城市平均收入 SELECT city, <u>AVG</u>(salary) FROM employee GROUP BY city; SELECT city, description, count(*) FROM employee <u>GROUP BY city, description</u>;</pre>	<pre>SELECT city, COUNT(*) FROM employee <u>GROUP BY city</u> ORDER BY COUNT(*); select city, avg(salary) avg_salary from employee group by city order by avg(salary) desc; //各城市平均收入 (只選 > 3000) SELECT city, <u>AVG</u>(salary) FROM employee WHERE salary < 5000 GROUP BY city <u>HAVING AVG</u>(salary) > 3000 ORDER BY <u>AVG</u>(salary); drop table temp; // 各城市人數 2,1,5 create table <u>temp</u> as SELECT <u>count</u>(*) as cnt FROM employee GROUP BY city;</pre>
--	--

<pre>SELECT count(*), city FROM employee GROUP BY city HAVING <u>count(*)</u>= (select <u>max(cnt)</u> from <u>temp</u>); // 最多</pre>	<pre>SELECT city, AVG(salary) FROM employee GROUP BY city HAVING AVG(salary) > 5000;</pre>
---	---

```
SELECT id, first_name, salary FROM employee
SELECT id, last_name FROM employee WHERE first_name IS NULL;
```

```
// pattern matching (%)0,1,2,...n, (_)1 char
SELECT * FROM employee WHERE first_name LIKE '%JA%'; // 包含 "JA"
SELECT * FROM employee WHERE first_name LIKE '%\_ %'; // 包含 "_"
SELECT * FROM employee WHERE first_name LIKE 'J%'; // 第 1 字符"J"
SELECT * FROM employee WHERE first_name LIKE '?ason'; // "?ason"
SELECT * FROM employee WHERE first_name LIKE ' o%'; // 第 2 字符"o"
SELECT * FROM employee WHERE id NOT BETWEEN '01' AND '03';
```

```
SELECT id, first_name, salary FROM employee ORDER BY salary
SELECT * FROM employee ORDER BY start_date, first_name;
SELECT * FROM employee ORDER BY last_name;
SELECT * FROM employee ORDER BY last_name Desc;
SELECT * FROM employee ORDER BY first_name ASC, last_name DESC;
SELECT id, first_name, last_name FROM employee ORDER BY 2;
SELECT id, first_name, last_name FROM employee
SELECT id, salary, salary-80000, ABS(salary-80000) FROM employee;
```

```
select city, sum(salary), avg(salary), min(salary) , max(salary), count(*)
from employee group by city;
```

```
select * from employee where id=5 and salary>0
```

<pre>SELECT id, first_name, salary FROM employee WHERE salary < 43000 ORDER BY salary; SELECT * FROM employee WHERE salary < 50000; SELECT * FROM employee WHERE id = 1;</pre>	<pre>select first_name, salary from employee where (first_name = 'Jason' or salary <> 2000) and not (first_name = 'Jason' and salary <> 2000); select start_date from employee where start_date > '2000-1-31'</pre>
---	---

<pre>SELECT first_name FROM employee WHERE city = 'Vancouver';</pre>	<pre>select first_name from employee where id in (select id from employee where city = 'Vancouver');</pre>
--	--

<pre>CREATE TABLE timeTable (year YEAR(4), month INT(2) unsigned zerofill, day INT(2) unsigned zerofill);</pre>	<pre>INSERT INTO timeTable VALUES (2001,1,1), (2002,6,20), (2003,5,30), (2004,2,2), (2005,4,23), (2006,3,23);</pre>
--	---

year	month	day
2001	01	01
2002	06	20
2003	05	30
2004	02	02
2005	04	23
2006	03	23

```
select * from timeTable;
select * from TimeTable WHERE year BETWEEN 2001 AND 2003;
```

```
Drop table employee0;
CREATE TABLE employee0 (
id          int unsigned not null auto_increment primary key,
firstname   VARCHAR(20),
lastname    VARCHAR(20),
title       VARCHAR(30),
age         int,
yearofservice int,
salary      int,
bonus       int,
email       VARCHAR(60)
);
```

id	firstname	lastname	title	age	Yearof service	salary	bonus	email
1	John	Chen	Senior Programmer	31	3	120000	25000	j@hotmail.com
2	Jan	Pillai	Senior Programmer	32	4	110000	20000	g@yahoo.com
3	Ane	Pandit	Web Designer	24	3	90000	15000	a@gmail.com
4	Mary	Anchor	Web Designer	27	2	85000	15000	m@mail.com
5	Fred	King	Programmer	32	3	75000	15000	f@net.com
6	John	Mac	Programmer	32	4	80000	16000	j@hotmail.com
7	Arthur	Sam	Programmer	28	2	75000	14000	e@yahoo.com
8	Alok	Nanda	Programmer	32	3	70000	10000	a@yahoo.com
9	Susan	Ra	Multimedia Programmer	32	4	90000	15000	h@gmail.com
10	Paul	Simon	Multimedia Programmer	23	1	85000	12000	ps@gmail.com
11	Edward	Parhar	Multimedia Programmer	30	2	75000	15000	a@hotmail.com
12	Kim	Hunter	Senior Web Designer	32	4	110000	20000	kim@coolmail.com
13	Roger	Lewis	System Administrator	32	3	100000	13000	roger@mail.com
14	Danny	Gibson	System Administrator	31	2	90000	12000	danny@hotmail.com
15	Mike	Harper	Senior Marketing Executive	36	1	120000	28000	m@gmail.com
16	Mary	Sunday	Marketing Executive	31	5	90000	25000	monica@bigmail.com
17	Jack	Sim	Marketing Executive	27	1	70000	18000	hal@gmail.com
18	Joe	Irvine	Marketing Executive	27	1	72000	18000	joseph@hotmail.com
19	Henry	Ali	Customer Service Manager	32	3	70000	9000	shahida@hotmail.com
20	Peter	Champion	Finance Manager	32	2	120000	25000	peter@yahoo.com

```

INSERT INTO employee0 (firstname,lastname,title,age,yearofservice,salary,bonus,email)
values ("John", "Chen", "Senior Programmer", 31, 3, 120000, 25000, "j@hotmail.com");
INSERT INTO employee0 (firstname,lastname,title,age,yearofservice,salary,bonus,email)
values ("Jan", "Pillai", "Senior Programmer", 32, 4, 110000, 20000, "g@yahoo.com");
INSERT INTO employee0 (firstname,lastname,title,age,yearofservice,salary,bonus,email)
values ("Ane", "Pandit", "Web Designer", 24, 3, 90000, 15000, "a@gmail.com");
INSERT INTO employee0 (firstname,lastname,title,age,yearofservice,salary,bonus,email)
values ("Mary", "Anchor", "Web Designer", 27, 2, 85000, 15000, "m@mail.com");
INSERT INTO employee0 (firstname,lastname,title,age,yearofservice,salary,bonus,email)
values ("Fred", "King", "Programmer", 32, 3, 75000, 15000, "f@net.com");
INSERT INTO employee0 (firstname,lastname,title,age,yearofservice,salary,bonus,email)
values ("John", "Mac", "Programmer", 32, 4, 80000, 16000, "j@hotmail.com");
INSERT INTO employee0 (firstname,lastname,title,age,yearofservice,salary,bonus,email)
values ("Arthur", "Sam", "Programmer", 28, 2, 75000, 14000, "e@yahoo.com");
INSERT INTO employee0 (firstname,lastname,title,age,yearofservice,salary,bonus,email)
values ("Alok", "Nanda", "Programmer", 32, 3, 70000, 10000, "a@yahoo.com");
INSERT INTO employee0 (firstname,lastname,title,age,yearofservice,salary,bonus,email)
values ("Susan", "Ra", "Multimedia Programmer", 32, 4, 90000, 15000, "h@gmail.com");
INSERT INTO employee0 (firstname,lastname,title,age,yearofservice,salary,bonus,email)
values ("Paul", "Simon", "Multimedia Programmer", 23, 1, 85000, 12000, "ps@gmail.com");
INSERT INTO employee0 (firstname,lastname,title,age,yearofservice,salary,bonus,email)
values ("Edward", "Parhar", "Multimedia Programmer", 30, 2, 75000, 15000, "a@hotmail.com");
INSERT INTO employee0 (firstname,lastname,title,age,yearofservice,salary,bonus,email)
values ("Kim", "Hunter", "Senior Web Designer", 32, 4, 110000, 20000, "kim@coolmail.com");
INSERT INTO employee0 (firstname,lastname,title,age,yearofservice,salary,bonus,email)
values ("Roger", "Lewis", "System Administrator", 32, 3, 100000, 13000, "roger@mail.com");
INSERT INTO employee0 (firstname,lastname,title,age,yearofservice,salary,bonus,email)
values ("Danny", "Gibson", "System Administrator", 31, 2, 90000, 12000, "danny@hotmail.com");
INSERT INTO employee0 (firstname,lastname,title,age,yearofservice,salary,bonus,email)
values ("Mike", "Harper", "Senior Marketing Executive", 36, 1, 120000, 28000, "m@gmail.com");
INSERT INTO employee0 (firstname,lastname,title,age,yearofservice,salary,bonus,email)
values ("Mary", "Sunday", "Marketing Executive", 31, 5, 90000, 25000, "monica@bigmail.com");
INSERT INTO employee0 (firstname,lastname,title,age,yearofservice,salary,bonus,email)
values ("Jack", "Sim", "Marketing Executive", 27, 1, 70000, 18000, "hal@gmail.com");
INSERT INTO employee0 (firstname,lastname,title,age,yearofservice,salary,bonus,email)
values ("Joe", "Irvine", "Marketing Executive", 27, 1, 72000, 18000, "joseph@hotmail.com");
INSERT INTO employee0 (firstname,lastname,title,age,yearofservice,salary,bonus,email)
values ("Henry", "Ali", "Customer Service Manager", 32, 3, 70000, 9000, "shahida@hotmail.com");
INSERT INTO employee0 (firstname,lastname,title,age,yearofservice,salary,bonus,email)
values ("Peter", "Champion", "Finance Manager", 32, 2, 120000, 25000, "peter@yahoo.com");

```

```
SELECT * FROM employee0;
```

```
SELECT firstname, lastName, yearofservice  
FROM employee0  
ORDER BY yearofservice;
```

```
// new tables:
```

```
CREATE TABLE Employee2(  
empno          int(3),  
ename          VARCHAR(15),  
hiredate       DATE,  
orig_salary    int(6),  
curr_salary    int(6),  
region         VARCHAR(1)  
); // (員工編號,姓名,顧用日期,原來薪金,現時薪金,地區)
```

EMPNO	ENAME	HIREDATE	ORIG_SALARY	CURR_SALARY	REGION
1	Jason	1996-07-25	1234	8767	E
2	John	1997-07-15	2341	3456	W
3	Joe	1986-01-25	4321	5654	E
4	Tom	2006-09-13	2413	6787	W
5	Jane	2005-04-17	7654	4345	E
6	James	2004-07-18	5679	6546	W
7	Jodd	2003-07-20	5438	7658	E
8	Joke	2002-01-01	8765	4543	W
9	Jack	2001-08-29	7896	1232	E

```
insert into Employee2 values
```

```
(1, 'Jason','19960725', 1234, 8767, 'E');  
(2, 'John','19970715', 2341, 3456, 'W');  
(3, 'Joe', '19860125', 4321, 5654, 'E');  
(4, 'Tom', '20060913', 2413, 6787, 'W');  
(5, 'Jane','20050417', 7654, 4345, 'E');  
(6, 'James','20040718', 5679, 6546, 'W');  
(7, 'Jodd','20030720', 5438, 7658, 'E');  
(8, 'Joke','20020101', 8765, 4543, 'W');  
(9, 'Jack','20010829', 7896, 1232, 'E');
```

```
SELECT * FROM employee2;  
SELECT * FROM employee2 ORDER BY orig_salary;
```



```
SELECT * FROM employee2
WHERE hiredate BETWEEN '20000415' and '20030212';
```

```
// new tables
```

```
CREATE TABLE Employee3(
empno          int(3),
ename          VARCHAR(15),
hiredate       DATE,
orig_salary    int(6),
curr_salary    int(6),
region         VARCHAR(1),
manager_id     int(3)
); // (員工編號,姓名,顧用日期,原來薪金,現時薪金,地區,上司)
```

EMPNO	ENAME	HIREDATE	ORIG_SALARY	CURR_SALARY	REGION	MANAGER_ID
1	Jason	1996-07-25	1234	8767	E	2
2	John	1997-07-15	2341	3456	W	3
3	Joe	1986-01-25	4321	5654	E	3
4	Tom	2006-09-13	2413	6787	W	4
5	Jane	2005-04-17	7654	4345	E	4
6	James	2004-07-18	5679	6546	W	5
7	Jodd	2003-07-20	5438	7658	E	6
8	Joke	2002-01-01	8765	4543	W	NULL
9	Jack	2001-08-29	7896	1232	E	NULL

```
insert into Employee3 values (1, 'Jason','19960725', 1234, 8767, 'E', 2);
insert into Employee3 values (2, 'John','19970715', 2341, 3456, 'W', 3);
insert into Employee3 values (3, 'Joe', '19860125', 4321, 5654, 'E', 3);
insert into Employee3 values (4, 'Tom','20060913', 2413, 6787, 'W', 4);
insert into Employee3 values (5, 'Jane','20050417', 7654, 4345, 'E', 4);
insert into Employee3 values (6, 'James','20040718', 5679, 6546, 'W', 5);
insert into Employee3 values (7, 'Jodd','20030720', 5438, 7658, 'E', 6);
insert into Employee3 (EMPNO,ENAME,HIREDATE,ORIG_SALARY,CURR_SALARY,REGION)
values (8, 'Joke','20020101', 8765, 4543, 'W');
insert into Employee3 (EMPNO,ENAME,HIREDATE,ORIG_SALARY,CURR_SALARY,REGION)
values (9, 'Jack','20010829', 7896, 1232, 'E'); // 沒有 manager_id
```

```
CREATE TABLE job (
empno int(3),
jobtitle VARCHAR(20)
); // (員工編號,職位/工作崗位)
```

EMPNO	jobtitle
1	Tester
2	Accountant
3	Developer
4	Coder
5	Director
6	Mediator
7	Professor
8	Programmer
9	Developer

```

insert into job (EMPNO, Jobtitle) values (1,'Tester');
insert into job (EMPNO, Jobtitle) values (2,'Accountant');
insert into job (EMPNO, Jobtitle) values (3,'Developer');
insert into job (EMPNO, Jobtitle) values (4,'Coder');
insert into job (EMPNO, Jobtitle) values (5,'Director');
insert into job (EMPNO, Jobtitle) values (6,'Mediator');
insert into job (EMPNO, Jobtitle) values (7,'Professor');
insert into job (EMPNO, Jobtitle) values (8,'Programmer');
insert into job (EMPNO, Jobtitle) values (9,'Developer');

```

<pre> CREATE TABLE Professor (ProfessorID INT NOT NULL PRIMARY KEY, Name VARCHAR(50) NOT NULL); </pre>	<pre> CREATE TABLE Student (StudentID INT NOT NULL PRIMARY KEY, Name VARCHAR(50) NOT NULL); </pre>
---	---

StudentID	Name
1	Joe Wang
2	Henry Al
3	Amma Zee
4	Lili Lee
5	Sam Jun
6	Dianna Wang

```

INSERT INTO Student (StudentID,Name) VALUES (1,'John Jones');
INSERT INTO Student (StudentID,Name) VALUES (2,'Cury Butz');
INSERT INTO Student (StudentID,Name) VALUES (3,'JJ Smith');

```

// 複製資料 student → professor

```

INSERT INTO Professor (ProfessorID, Name)
SELECT StudentID + 7, Name FROM Student;

```

```

CREATE TABLE CDs (
  CDID          SMALLINT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,
  CDName        VARCHAR(50) NOT NULL,
  Copyright     YEAR,
  NumberDisks   TINYINT UNSIGNED NOT NULL DEFAULT 1,
  NumberInStock TINYINT UNSIGNED,
  NumberOnReserve TINYINT UNSIGNED NOT NULL,
  NumberAvailable TINYINT UNSIGNED NOT NULL,
  CDType        VARCHAR(20),
  RowAdded      TIMESTAMP
);

```

```

INSERT INTO CDs VALUES

```

```

  (NULL, 'Earle', 1996, 2, 10, 3, NumberInStock-NumberOnReserve, 'Country', NULL);

```

```

// cd ID, name, 版權, 碟, 存貨量, 保留/預訂數量, 真實數量, 分類, 輸入日期

```

CDID	CDName	Copyright	Number Disks	Number InStock	Number OnReserve	Number Available	CDType 分類	RowAdded
1	Earle	1996	2	10	3	7	Country	2011-04-27
2	Variations	1999	1	9	0	9	Blues	2011-04-27
3	The	1990	1	14	2	12	Popular	2011-04-27
4	Shocked	1988	1	6	1	5	Folk-Rock	2011-04-27

```

INSERT INTO CDs (CDName, Copyright, NumberDisks, NumberInStock, NumberOnReserve,
NumberAvailable, CDType)

```

```

VALUES ('Variations', 1999, 1, 9, 0, NumberInStock-NumberOnReserve, 'Blues'),
  ('The', 1990, 1, 14, 2, NumberInStock-NumberOnReserve, 'Popular'),
  ('Shocked', 1988, 1, 6, 1, NumberInStock-NumberOnReserve, 'Folk-Rock');

```

```

// regular expression 萬用字元

```

```

SELECT * FROM employee WHERE name REGEXP '^.....$'; // exactly 5 char

```

```

CREATE TABLE Sale(

```

```

  ID          SMALLINT NOT NULL AUTO_INCREMENT PRIMARY KEY,
  Name        VARCHAR(50) NOT NULL,
  InStock     SMALLINT UNSIGNED NOT NULL,
  OnOrder     SMALLINT UNSIGNED NOT NULL,
  Reserved    SMALLINT UNSIGNED NOT NULL,
  Department  ENUM('Classical', 'Popular') NOT NULL,
  Category    VARCHAR(20)
);

```

ID	Name	InStock	OnOrder	Reserved	Department 部門	Category 分類
1	Bloodshot	11	6	1	Popular	Rock
2	Opera	12	5	2	Classical	Opera
3	Jazz	13	4	3	Popular	Jazz
4	Music	4	3	4	Classical	Dance
5	Violin	25	2	5	Classical	NULL
6	Toscana	16	1	6	Classical	NULL
7	Blues	7	22	7	Popular	Blues
8	Pure	38	5	11	Popular	NULL
9	Mud	19	11	12	Popular	Country
10	Essence	5	23	12	Popular	New Age
11	Embrace	21	12	14	Popular	New Age
12	Satie	42	17	15	Classical	NULL
13	Lake	23	47	28	Classical	Dance
14	Favorites	34	15	12	Classical	General
15	Boheme	25	12	5	Classical	Opera
16	Cantatas	26	13	8	Classical	General
17	Road	27	13	17	Popular	Country
18	Paris	18	25	10	Popular	Jazz
19	Woman	29	4	7	Popular	Blues
20	Bach	21	13	16	Classical	General
21	Opera	12	32	12	Classical	Opera
22	Soul	13	30	14	Popular	NULL
23	Stages	44	0	8	Popular	Blues
24	Bach	15	1	8	Classical	General

INSERT INTO Sale (Name, InStock, OnOrder, Reserved, Department, Category)

VALUES ('Bloodshot', 11, 6, 1, 'Popular', 'Rock'),

('Opera', 12, 5, 2, 'Classical', 'Opera'),

('Jazz', 13, 4, 3, 'Popular', 'Jazz'),

('Music', 4, 3, 4, 'Classical', 'Dance'),

('Violin', 25, 2, 5, 'Classical', NULL),

('Toscana', 16, 1, 6, 'Classical', NULL),

('Blues', 7, 22, 7, 'Popular', 'Blues'),

('Pure', 38, 5, 11, 'Popular', NULL),

('Mud', 19, 11, 12, 'Popular', 'Country'),

('Essence', 5, 23, 12, 'Popular', 'New Age'),

('Embrace', 21, 12, 14, 'Popular', 'New Age'),

('Satie', 42, 17, 15, 'Classical', NULL),

```

('Lake', 23, 47, 28, 'Classical', 'Dance'),
('Favorites', 34, 15, 12, 'Classical', 'General'),
('Boheme', 25, 12, 5, 'Classical', 'Opera'),
('Cantatas', 26, 13, 8, 'Classical', 'General'),
('Road', 27, 13, 17, 'Popular', 'Country'),
('Paris', 18, 25, 10, 'Popular', 'Jazz'),
('Woman', 29, 4, 7, 'Popular', 'Blues'),
('Bach', 21, 13, 16, 'Classical', 'General'),
('Opera', 12, 32, 12, 'Classical', 'Opera'),
('Soul', 13, 30, 14, 'Popular', NULL),
('Stages', 44, 0, 8, 'Popular', 'Blues'),
('Bach', 15, 1, 8, 'Classical', 'General');

```

// regular expression (^start, \$end) 萬用字元

```

SELECT Name, InStock
FROM Sale
WHERE Name REGEXP '^[mn].*[cd]$\''
ORDER BY Name;

```

<pre> CREATE TABLE Student (StudentID INT NOT NULL PRIMARY KEY, Name VARCHAR(50) NOT NULL); </pre>	<pre> CREATE TABLE StudentExam (StudentID INT NOT NULL, ExamID INT NOT NULL, Mark INT, IfPassed SMALLINT, Comments VARCHAR(255)); </pre>
--	---

```

CREATE TABLE Exam (
ExamID       INT NOT NULL PRIMARY KEY,
CourseID     INT NOT NULL,
ProfessorID  INT NOT NULL,
SustainedOn  DATE,
Comments     VARCHAR(255),
INDEX       examcourse_index(CourseID),
CONSTRAINT  FK_ExamCourse FOREIGN KEY (CourseID)
            REFERENCES Course(CourseID),
INDEX       examprof_index(ProfessorID),
CONSTRAINT  FK_ExamProf FOREIGN KEY (ProfessorID)
            REFERENCES Professor(ProfessorID)
);

```

StudentID	ExamID	Mark	IfPassed	Comments 備註/評語
1	1	55	1	Satisfactory
1	2	73	1	Good
2	3	44	1	Scraped through
2	5	39	0	Failed
2	6	63	1	NULL
3	4	78	1	Excellent
3	7	82	1	Great
4	8	65	1	Adequate
4	11	72	1	Good
4	5	53	1	Below expectations
5	1	26	0	Very poor
5	9	68	1	Good
5	2	62	1	Good
6	3	78	1	Excellent
6	6	69	1	NULL
6	10	58	1	Adequate performance

// 學生 ID, 姓名

```
INSERT INTO Student (StudentID,Name) VALUES (1,'Joe Wang');
INSERT INTO Student (StudentID,Name) VALUES (2,'Henry AI');
INSERT INTO Student (StudentID,Name) VALUES (3,'Amma Zee');
INSERT INTO Student (StudentID,Name) VALUES (4,'Lili Lee');
INSERT INTO Student (StudentID,Name) VALUES (5,'Sam Jun');
INSERT INTO Student (StudentID,Name) VALUES (6,'Dianna Wang');
```

// 考試 ID, 課程 ID, 教授 ID, 進行日期, 備註/評語

```
INSERT INTO Exam (ExamID,CourseID,ProfessorID,SustainedOn,Comments)
VALUES (1,1,1,'2003-03-12','A difficult test that should last an hour');
INSERT INTO Exam (ExamID,CourseID,ProfessorID,SustainedOn,Comments)
VALUES (2,2,1,'2003-03-13','A simple two hour test');
INSERT INTO Exam (ExamID,CourseID,ProfessorID,SustainedOn,Comments)
VALUES (3,3,2,'2003-03-11','1 hour long');
INSERT INTO Exam (ExamID,CourseID,ProfessorID,SustainedOn)
VALUES (4,4,3,'2003-03-18');
INSERT INTO Exam (ExamID,CourseID,ProfessorID,SustainedOn,Comments)
VALUES (5,5,2,'2003-03-19','2 hours long');
INSERT INTO Exam (ExamID,CourseID,ProfessorID,SustainedOn)
VALUES (6,6,3,'2003-03-25');
```

```
// 學生 ID, 考試 ID, 分數, pass/fail, 備註/評語
INSERT INTO StudentExam (StudentID,ExamID,Mark,IfPassed,Comments) VALUES
    (1,1,55,1,'Satisfactory'),
    (1,2,73,1,'Good result'),
    (2,3,44,1,'Scraped through'),
    (2,5,39,0,'Failed, and will need to retake this one later in the year'),
    (2,6,63,1, ''),
    (3,4,78,1,'Excellent result'),
    (3,7,82,1,'Great result!'),
    (4,8,65,1,'Adequate performance'),
    (4,11,72,1,'Good result'),
    (4,5,53,1,'Below expectations'),
    (5,1,26,0,'Very poor performance. Recommend to drop this module'),
    (5,9,68,1,'Good result'),
    (5,2,62,1,'Good result'),
    (6,3,78,1,'Excellent work'),
    (6,6,69,1, ''),
    (6,10,58,1,'Adequate performance');
```

<pre>// students who failed in exam 不合格 SELECT StudentID, Name FROM Student s WHERE EXISTS (SELECT StudentID FROM StudentExam e WHERE Mark < 40 AND e.StudentID = s.StudentID);</pre>	<p>不合格</p> <pre>SELECT s.StudentID, Name FROM Student s, StudentExam e WHERE Mark<40 AND e.StudentID = s.StudentID</pre>
---	---

<pre>// update records 更新紀錄 UPDATE employee SET salary = salary+1000 WHERE description='Programmer';</pre>	<pre>// delete records 刪除紀錄 DELETE FROM employee WHERE description='Programmer';</pre>
<pre>System date: sysdate() Current date: curdate() ... where instr(name,'son') ... where name LIKE '%son%' // name 包含 'son'</pre>	<pre>Left(name,5) Mid(name,2,5) Substr(name,2,5) Upper(name)</pre>

```
CREATE TABLE enrollment(
sid      int NOT NULL,
course   char(10) NOT NULL,
score    int
); // 註冊(學生編號,課程名稱,分數)
```

```
insert into enrollment
(sid,course) values
(101,"中文"),
(102,"電腦"),
(103,"會計"),
(104,"數學"),
(105,"電腦"),
(102,"會計"),
(103,"數學"),
(101,"英文"),
(106,"中文"),
(104,"電腦"),
(108,"會計"),
(109,"數學");
```

enrollment e1

sid	course	score
101	中文	
102	電腦	
103	會計	
104	數學	
105	電腦	
102	會計	
103	數學	
101	英文	
106	中文	
104	電腦	
108	會計	
109	數學	
...		

enrollment e2

sid	course	score
101	中文	
102	電腦	
103	會計	
104	數學	
105	電腦	
102	會計	
103	數學	
101	英文	
106	中文	
104	電腦	
108	會計	
109	數學	
...		

```
// 同時修讀電腦及數學
SELECT sid FROM enrollment
WHERE course = '電腦'
AND sid IN ( )
```

```
// 同時修讀電腦及數學
SELECT e1.sid
FROM enrollment e1, enrollment e2
WHERE e1.sid = e2.sid
AND e1.course = '電腦'
AND e2.course = '數學'
```

```
SELECT sid FROM enrollment
WHERE course = '數學'
```