Introduction to Databases

CT042-3-1-IDB (version1)

SQL

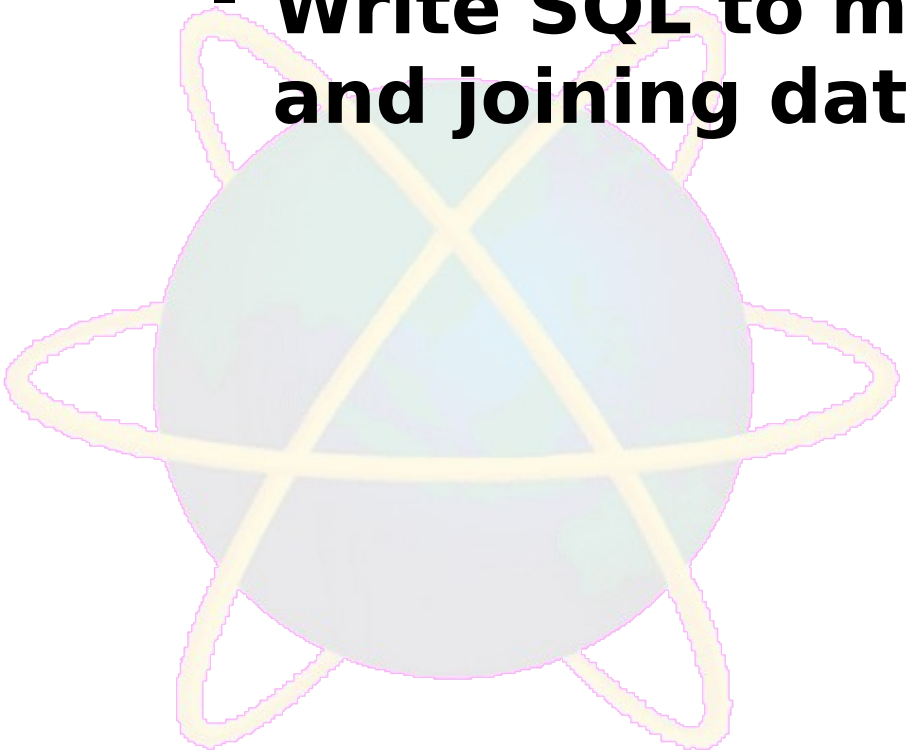
Instructor: Bidur Devkota

Topic & Structure of The Lesson

- Aggregate functions
- Altering data type
- Grouping data
- Joining database

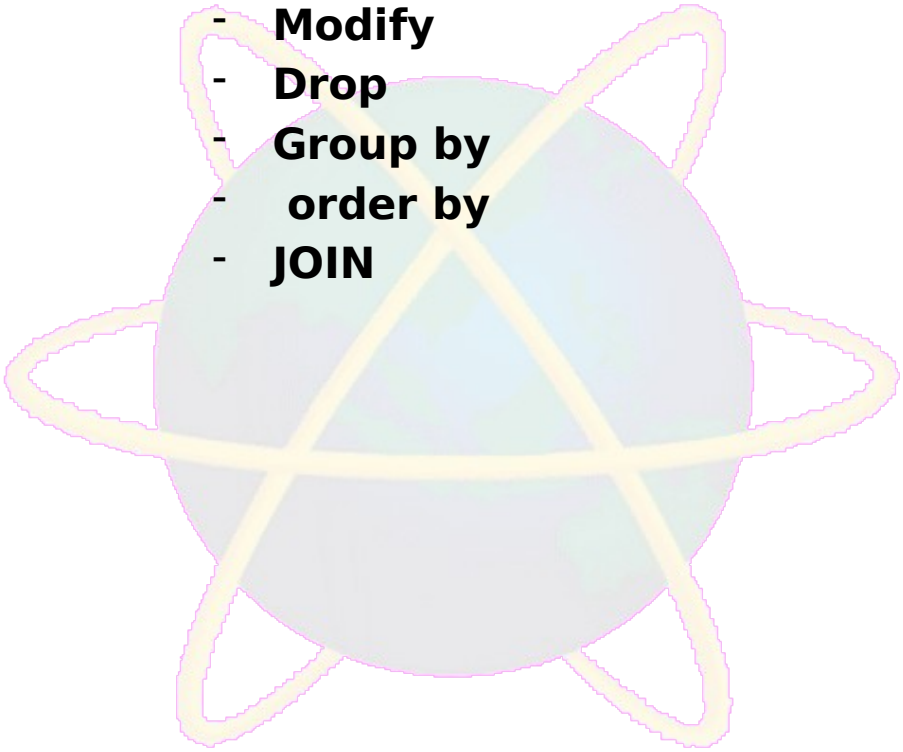
Learning Outcomes

- **At the end of this topic, You should be able to**
  - **Write SQL to manipulate database and joining database**



Key Terms You Must Be Able To Use

- **If you have mastered this topic, you should be able to use the following terms correctly in your assignments and exams:**
  - **Alter**
  - **Modify**
  - **Drop**
  - **Group by**
  - **order by**
  - **JOIN**



Advanced Data Definition Commands

- All changes in table structure are made by using ALTER command
  - Followed by keyword that produces specific change
  - Following three options are available:
    - ADD
    - MODIFY
    - DROP

Changing a Column's Data Type

- **ALTER** can be used to **change data type**
- Some RDBMSs (such as Oracle) do not permit changes to data types unless column to be changed is empty
- **MySQL Example**- change col name & data

```
ALTER TABLE students CHANGE sem semester INT;
```

Changing a Column's Data Characteristics

- Use ALTER to change data characteristics
- If column to be changed already contains data, **changes in column's characteristics are permitted if those changes do not alter the data type**

Adding a Column

- Use ALTER to add column
  - Do not include the NOT NULL clause for new column

```
ALTER TABLE students ADD department VARCHAR(20);
```

Dropping a Column

- Use ALTER to drop column
  - Some RDBMSs impose restrictions on the deletion of an attribute

```
mysql> desc students;
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| roll       | int(11)     | YES  |     | NULL    |       |
| name       | varchar(50) | YES  |     | NULL    |       |
| phone      | int(11)     | YES  |     | NULL    |       |
| sem        | varchar(20) | YES  |     | NULL    |       |
| department | varchar(20) | YES  |     | NULL    |       |
+------------+-------------+------+-----+---------+-------+
5 rows in set (0.00 sec)

mysql> ALTER TABLE students DROP COLUMN department;
Query OK, 0 rows affected (0.58 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> desc students;
+-------+-------------+------+-----+---------+-------+
| Field | Type        | Null | Key | Default | Extra |
+-------+-------------+------+-----+---------+-------+
| roll  | int(11)     | YES  |     | NULL    |       |
| name  | varchar(50) | YES  |     | NULL    |       |
| phone | int(11)     | YES  |     | NULL    |       |
| sem   | varchar(20) | YES  |     | NULL    |       |
+-------+-------------+------+-----+---------+-------+
4 rows in set (0.00 sec)
```

Copying Parts of Tables

- SQL permits copying contents of selected table columns so that the data need not be reentered manually into newly created table(s)
- First create the PART table structure
- Next add rows to new PART table using PRODUCT table rows

Copying Parts of Tables (continued)

Available tables in Database?

```
mysql> show tables;
+------------------+
| Tables_in_test_db |
+------------------+
| contacts         |
| new_students     |
| std_bak          |
| students         |
+------------------+
4 rows in set (0.00 sec)
```

Copy data from Table students to Table new_students?

```
mysql> insert into new_students ( select * from students where roll<5);
Query OK, 4 rows affected (0.05 sec)
Records: 4  Duplicates: 0  Warnings: 0

mysql> select * from new_students;
+------+---------+----------+-------+
| roll | name    | phone    | sem   |
+------+---------+----------+-------+
|    2 | Manoj   |  1665566 | 2nd   |
|    4 | Sandesh |  1445777 | 2nd   |
|    3 | Binod   |  1445566 | IV    |
|    1 | Ravi    | 61444333 | first |
+------+---------+----------+-------+
4 rows in set (0.00 sec)
```

Available Data in new_students?

```
mysql> select * from new_students;
Empty set (0.00 sec)
```

Adding Primary and Foreign Key Designations

- When table is copied, integrity rules do not copy, so primary and foreign keys need to be manually defined on new table
- User ALTER TABLE command
  - Syntax:
    - ALTER TABLE *tablename* ADD PRIMARY KEY(*fieldname*);
    - For foreign key, use FOREIGN KEY in place of PRIMARY KEY

Adding Primary and Foreign Key Designations


```
mysql> ALTER TABLE new_students ADD PRIMARY KEY (roll);
Query OK, 0 rows affected (0.66 sec)
Records: 0  Duplicates: 0  Warnings: 0
mysql> desc new_students;
+-------+-------------+------+-----+---------+-------+
| Field | Type        | Null | Key | Default | Extra |
+-------+-------------+------+-----+---------+-------+
| roll  | int(11)     | NO   | PRI | NULL    |       |
| name  | varchar(50) | YES  |     | NULL    |       |
| sem   | varchar(20) | YES  |     | NULL    |       |
+-------+-------------+------+-----+---------+-------+
3 rows in set (0.00 sec)
```

```
mysql> ALTER TABLE contacts ADD PRIMARY KEY (id);
Query OK, 0 rows affected (0.69 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> desc contacts;
+---------+-------------+------+-----+---------+-------+
| Field   | Type        | Null | Key | Default | Extra |
+---------+-------------+------+-----+---------+-------+
| roll    | int(11)     | YES  |     | NULL    |       |
| phone   | int(11)     | YES  |     | NULL    |       |
| address | varchar(30) | YES  |     | NULL    |       |
| id      | int(11)     | NO   | PRI | NULL    |       |
+---------+-------------+------+-----+---------+-------+
4 rows in set (0.00 sec)
```

```
mysql> ALTER TABLE contacts ADD FOREIGN KEY (roll) REFERENCES new_students(roll);
Query OK, 0 rows affected (0.81 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> desc contacts;
+---------+-------------+------+-----+---------+-------+
| Field   | Type        | Null | Key | Default | Extra |
+---------+-------------+------+-----+---------+-------+
| roll    | int(11)     | YES  | MUL | NULL    |       |
| phone   | int(11)     | YES  |     | NULL    |       |
| address | varchar(30) | YES  |     | NULL    |       |
| id      | int(11)     | NO   | PRI | NULL    |       |
+---------+-------------+------+-----+---------+-------+
4 rows in set (0.00 sec)
```

Adding Primary and Foreign Key Designations

```
mysql> select * from new_students;
+------+---------+-------+
| roll | name    | sem   |
+------+---------+-------+
|    1 | Ravi    | first |
|    2 | Manoj   | 2nd   |
|    3 | Binod   | IV    |
|    4 | Sandesh | 2nd   |
+------+---------+-------+
4 rows in set (0.00 sec)
```

```
mysql> select * from contacts;
+------+---------+---------+----+
| roll | phone   | address | id |
+------+---------+---------+----+
|    1 |  234556 | PKR     |  1 |
|    2 | 1234556 | KTM     |  2 |
|    3 |    6666 | Dharan  |  3 |
+------+---------+---------+----+
3 rows in set (0.00 sec)
```

Primary Key constraint

```
mysql> insert into new_students values (4, "sita", '2nd');
ERROR 1062 (23000): Duplicate entry '4' for key 'PRIMARY'
```

Foreign Key constraint

```
mysql> insert into contacts values (44, 0111111,"Damak", 4);
ERROR 1452 (23000): Cannot add or update a child row: a foreign key constraint fails
 (`test_db`.`contacts`, CONSTRAINT `contacts_ibfk_1` FOREIGN KEY (`roll`) REFERENCES
 `new_students` (`roll`))
```

Adding Primary and Foreign Key Designations

```
mysql> desc students;
+--------+-------------+------+-----+---------+-------+
| Field  | Type        | Null | Key | Default | Extra |
+--------+-------------+------+-----+---------+-------+
| roll   | int(11)     | YES  |     | NULL    |       |
| name   | varchar(50) | YES  |     | NULL    |       |
| sem    | varchar(20) | YES  |     | NULL    |       |
| gender | varchar(11) | YES  |     | NULL    |       |
+--------+-------------+------+-----+---------+-------+
4 rows in set (0.00 sec)

mysql> alter table students add primary key (roll);
ERROR 1062 (23000): Duplicate entry '2' for key 'PRIMARY'
mysql> delete from students where roll =2;
Query OK, 2 rows affected (0.04 sec)
```

Column with duplicate value cannot be made primary key

So REMOVE duplicate value and then retry.

Now add primary key

```
mysql> alter table students add primary key (roll);
Query OK, 0 rows affected (0.74 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> desc students;
+--------+-------------+------+-----+---------+-------+
| Field  | Type        | Null | Key | Default | Extra |
+--------+-------------+------+-----+---------+-------+
| roll   | int(11)     | NO   | PRI | NULL    |       |
| name   | varchar(50) | YES  |     | NULL    |       |
| sem    | varchar(20) | YES  |     | NULL    |       |
| gender | varchar(11) | YES  |     | NULL    |       |
+--------+-------------+------+-----+---------+-------+
4 rows in set (0.00 sec)
```

Deleting a Table from the Database

- DROP
  - Deletes table from database
  - Syntax:
    - DROP TABLE *tablename*;

```
mysql> show tables;
+-------------------+
| Tables_in_infomax |
+-------------------+
| students          |
+-------------------+
1 row in set (0.01 sec)

mysql> DROP TABLE students;
Query OK, 0 rows affected (0.18 sec)

mysql> show tables;
Empty set (0.00 sec)
```

Advanced Select Queries

- SQL provides useful functions that can:
  - Count
  - Find minimum and maximum values
  - Calculate averages
- SQL allows user to limit queries to only those entries having no duplicates or entries whose duplicates may be grouped

COUNT, MIN, MAX, AVG

```
mysql> select * from students;
+------+----------+----------+-------+
| roll | name     | phone    | sem   |
+------+----------+----------+-------+
|    2 | Manoj    |  1665566 | 2nd   |
|    4 | Sandesh  |  1445777 | 2nd   |
|    3 | Binod    |  1445566 | IV    |
|    1 | Ravi     | 61444333 | first |
|    9 | Pratyush | 61444333 | first |
|    5 | Laxman   | 61444555 | 3rd   |
|    6 | Jerry    | 61444555 | 3rd   |
|   10 | Tom      |   999999 | 2     |
|   10 | Pratyush |   999999 | 2     |
|   10 | Prayash  |   999999 | 2     |
+------+----------+----------+-------+
10 rows in set (0.01 sec)
```

```
mysql> SELECT COUNT(roll) , max(roll), min(roll), avg(roll) , sum(roll) from students;
+-------------+-----------+-----------+-----------+-----------+
| COUNT(roll) | max(roll) | min(roll) | avg(roll) | sum(roll) |
+-------------+-----------+-----------+-----------+-----------+
|          10 |        10 |         1 |    6.0000 |        60 |
+-------------+-----------+-----------+-----------+-----------+
1 row in set (0.00 sec)
```

Ordering a Listing

```
mysql> select * from students ORDER BY name;
+------+----------+----------+-------+
| roll | name     | phone    | sem   |
+------+----------+----------+-------+
|    3 | Binod    |  1445566 | IV    |
|    6 | Jerry    | 61444555 | 3rd   |
|    5 | Laxman   | 61444555 | 3rd   |
|    2 | Manoj    |  1665566 | 2nd   |
|    9 | Pratyush | 61444333 | first |
|   10 | Pratyush |   999999 | 2     |
|   10 | Prayash  |   999999 | 2     |
|    1 | Ravi     | 61444333 | first |
|    4 | Sandesh  |  1445777 | 2nd   |
|   10 | Tom      |   999999 | 2     |
+------+----------+----------+-------+
10 rows in set (0.00 sec)

mysql> select * from students ORDER BY name DESC;
+------+----------+----------+-------+
| roll | name     | phone    | sem   |
+------+----------+----------+-------+
|   10 | Tom      |   999999 | 2     |
|    4 | Sandesh  |  1445777 | 2nd   |
|    1 | Ravi     | 61444333 | first |
|   10 | Prayash  |   999999 | 2     |
|    9 | Pratyush | 61444333 | first |
|   10 | Pratyush |   999999 | 2     |
|    2 | Manoj    |  1665566 | 2nd   |
|    5 | Laxman   | 61444555 | 3rd   |
|    6 | Jerry    | 61444555 | 3rd   |
|    3 | Binod    |  1445566 | IV    |
+------+----------+----------+-------+
10 rows in set (0.00 sec)
```

FIGURE 7.17 Selected PRODUCT table attributes: ordered by (ascending) P_PRICE

P_CODE	P_DESCRIPT	P_INDATE	P_PRICE
54778-2T	Rat-tail file, 1/8-in. fine	15-Dec-05	4.99
PVC23DRT	PVC pipe, 3.5-in., 8-ft.	20-Feb-06	5.87
SM-18277	1.25-in. metal screw, 25	01-Mar-06	6.99
SW-23116	2.5-in. wd. screw, 50	24-Feb-06	8.45
23109-HB	Claw hammer	20-Jan-06	9.95
23114-AA	Sledge hammer, 12 lb.	02-Jan-06	14.40
13-Q2/P2	7.25-in. pwr. saw blade	13-Dec-05	14.99
14-Q1/L3	9.00-in. pwr. saw blade	13-Nov-05	17.49
2238/QPD	B&D cordless drill, 1/2-in.	20-Jan-06	38.95
1546-QQ2	Hrd. cloth, 1/4-in., 2x50	15-Jan-06	39.95
1558-QW1	Hrd. cloth, 1/2-in., 3x50	15-Jan-06	43.99
2232/QWE	B&D jigsaw, 8-in. blade	24-Dec-05	99.87
2232/QTY	B&D jigsaw, 12-in. blade	30-Dec-05	109.92
11QER/31	Power painter, 15 psi., 3-nozzle	03-Nov-05	109.99
WR3/TT3	Steel matting, 4'x8'x1/6", .5" mesh	17-Jan-06	119.95
89-WRE-Q	Hicut chain saw, 16 in.	07-Feb-06	256.99

Ordering a Listing (continued)

FIGURE 7.18 **Telephone list query results**

	EMP_LNAME	EMP_FNAME	EMP_INITIAL	EMP_AREACODE	EMP_PHONE
▶	Brandon	Marie	G	901	882-0845
	Diante	Jorge	D	615	890-4567
	Genkazi	Leighla	W	901	569-0093
	Johnson	Edward	E	615	898-4387
	Jones	Anne	M	615	898-3456
	Kolmycz	George	D	615	324-5456
	Lange	John	P	901	504-4430
	Lewis	Rhonda	G	615	324-4472
	Saranda	Hermine	R	615	324-5505
	Smith	George	A	615	890-2984
	Smith	George	K	901	504-3339
	Smith	Jeanine	K	615	324-7883
	Smythe	Melanie	P	615	324-9006
	Vandam	Rhett		901	675-8993
	Washington	Rupert	E	615	890-4925
	Wiesenbach	Paul	R	615	897-4358
	Williams	Robert	D	615	890-3220

Ordering a Listing (continued)

```
mysql> select * from students ORDER BY phone,name DESC;
+------+----------+----------+-------+
| roll | name     | phone    | sem   |
+------+----------+----------+-------+
|   10 | Tom      |   999999 | 2     |
|   10 | Prayash  |   999999 | 2     |
|   10 | Pratyush |   999999 | 2     |
|    3 | Binod    |  1445566 | IV    |
|    4 | Sandesh  |  1445777 | 2nd   |
|    2 | Manoj    |  1665566 | 2nd   |
|    1 | Ravi     | 61444333 | first |
|    9 | Pratyush | 61444333 | first |
|    5 | Laxman   | 61444555 | 3rd   |
|    6 | Jerry    | 61444555 | 3rd   |
+------+----------+----------+-------+
10 rows in set (0.00 sec)
```

FIGURE 7.19 A query based on multiple restrictions

	P_DESCRIPT	V_CODE	P_INDATE	P_PRICE
▶	Sledge hammer, 12 lb.		02-Jan-06	14.40
	Claw hammer	21225	20-Jan-06	9.95
	9.00-in. pwr. saw blade	21344	13-Nov-05	17.49
	7.25-in. pwr. saw blade	21344	13-Dec-05	14.99
	Rat-tail file, 1/8-in. fine	21344	15-Dec-05	4.99
	Hrd. cloth, 1/2-in., 3x50	23119	15-Jan-06	43.99
	Hrd. cloth, 1/4-in., 2x50	23119	15-Jan-06	39.95
	B&D cordless drill, 1/2-in.	25595	20-Jan-06	38.95

Listing Unique Values

```
mysql> SELECT DISTINCT sem from students;
+--------+
| sem    |
+--------+
| 2nd    |
| IV     |
| first  |
| 3rd    |
| 2      |
+--------+
5 rows in set (0.00 sec)
```

FIGURE 7.20 A listing of distinct (different) V_CODE values in the PRODUCT table

V_CODE
21225
21231
21344
23119
24288
25595

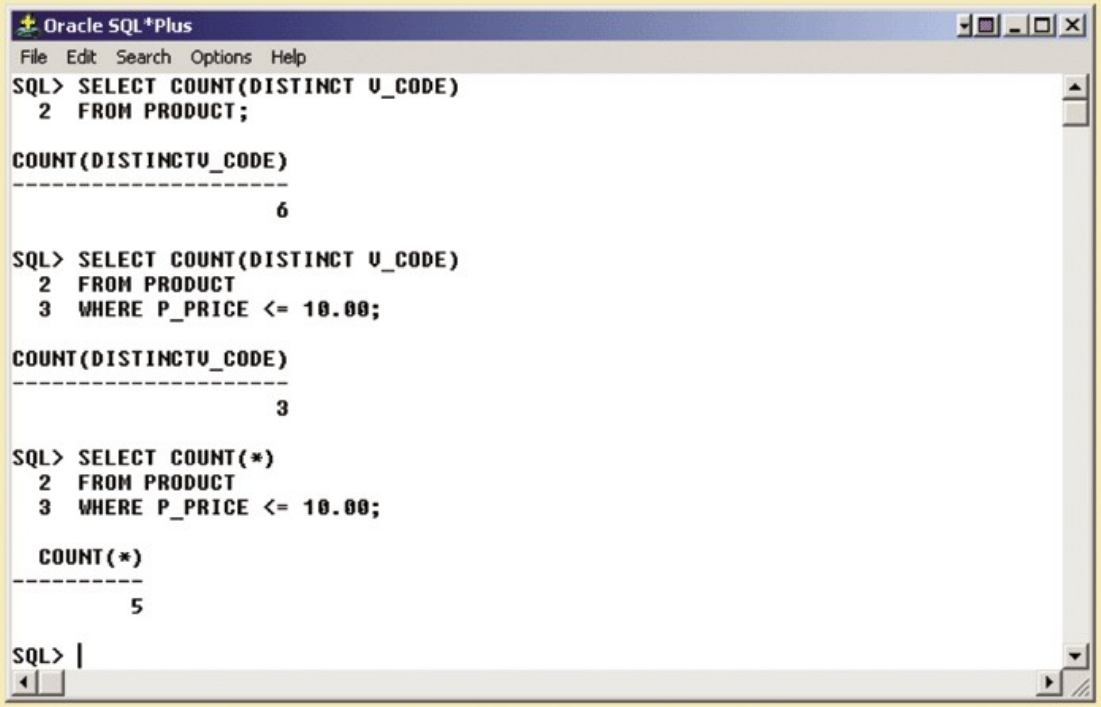
Aggregate Functions

TABLE 7.8 Some Basic SQL Aggregate Functions

FUNCTION	OUTPUT
COUNT	The number of rows containing non-null values
MIN	The minimum attribute value encountered in a given column
MAX	The maximum attribute value encountered in a given column
SUM	The sum of all values for a given column
AVG	The arithmetic mean (average) for a specified column

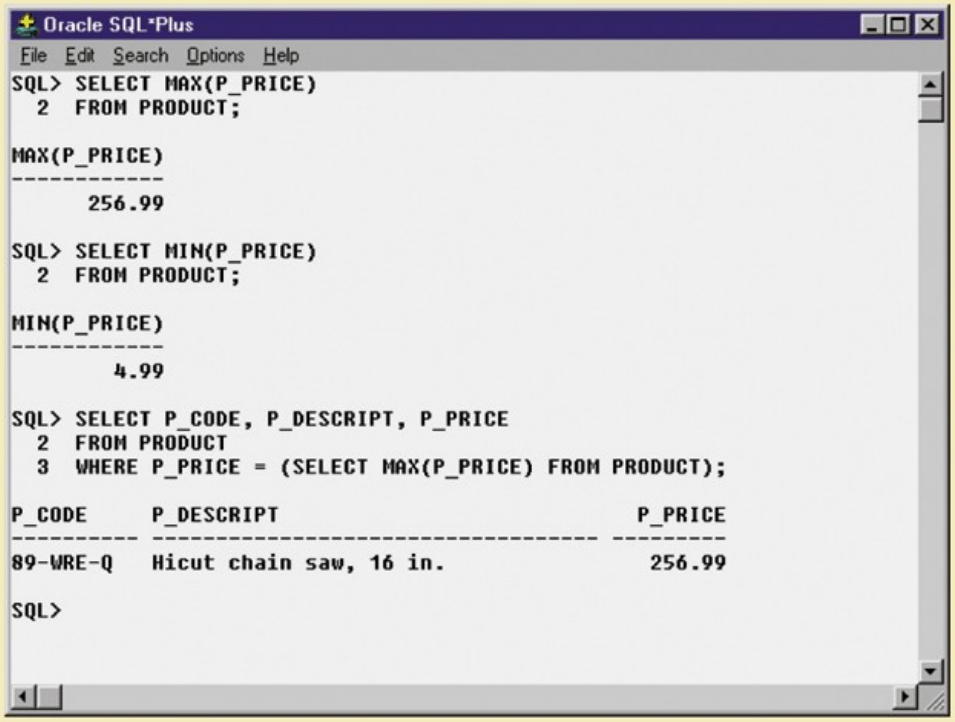
Aggregate Functions (continued)

FIGURE 7.21 COUNT function output examples



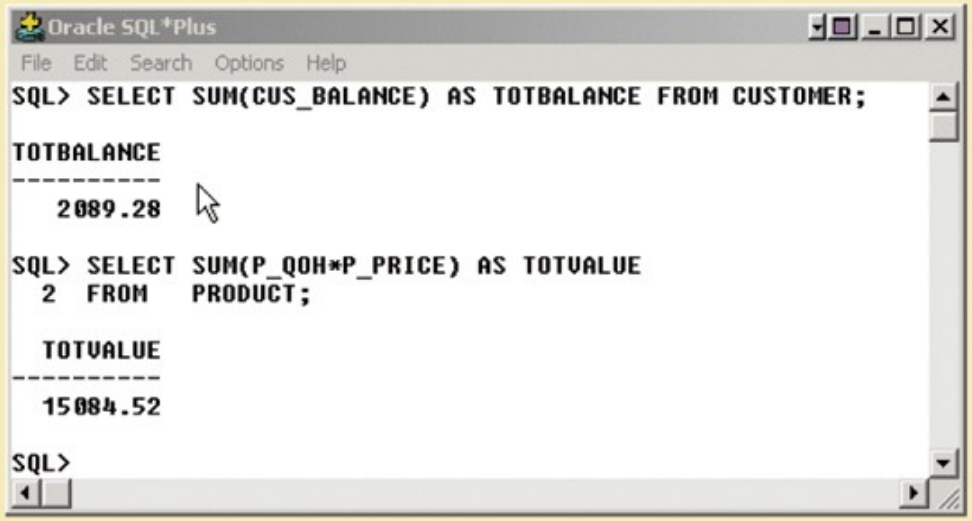
Aggregate Functions (continued)

FIGURE 7.22 MAX and MIN function output examples



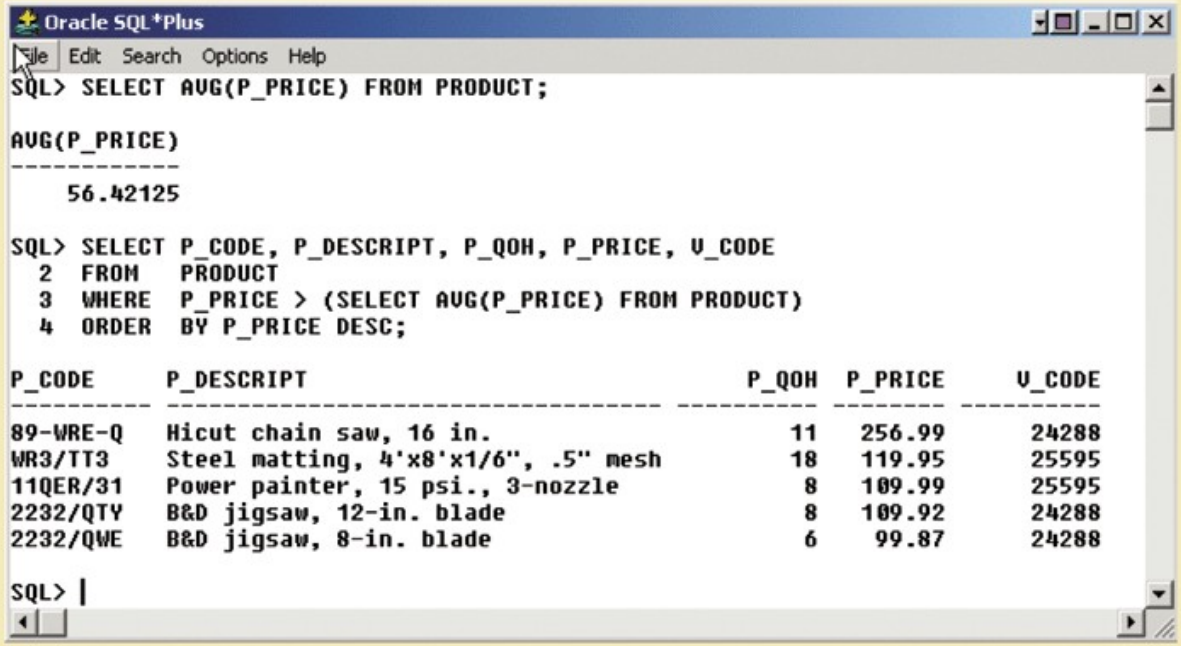
Aggregate Functions (continued)

FIGURE 7.23 The total value of all items in the PRODUCT table



Aggregate Functions (continued)

FIGURE 7.24 AVG function output examples



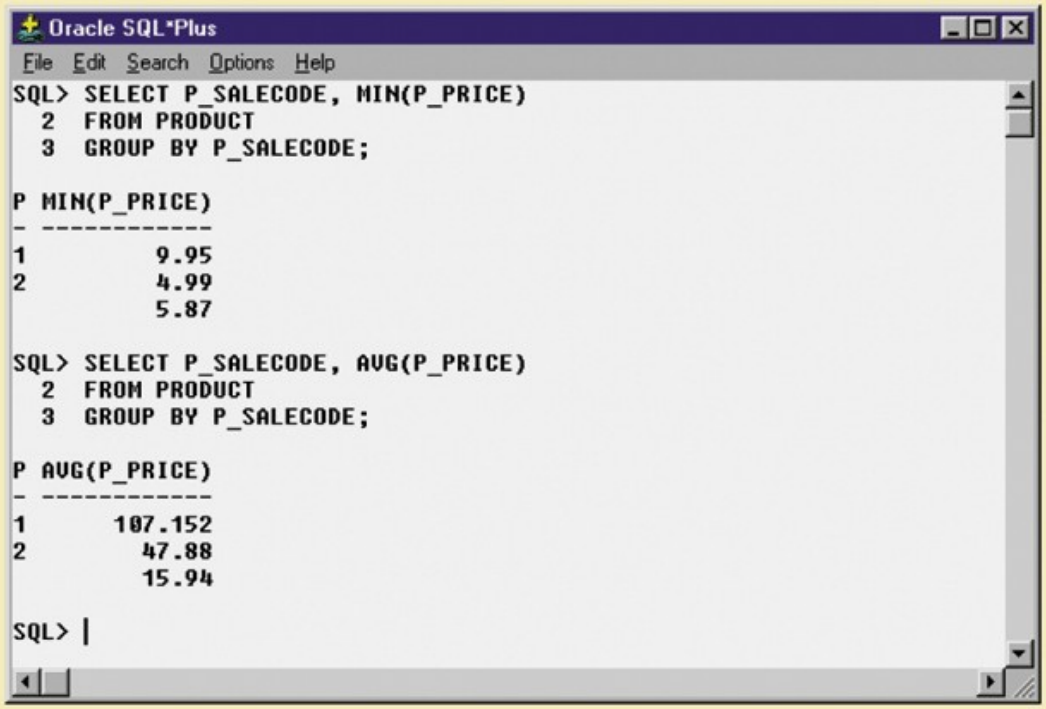
Grouping Data

```
mysql> select * from students;
+------+----------+----------+-------+--------+
| roll | name     | phone    | sem   | gender |
+------+----------+----------+-------+--------+
|    2 | Manoj    |  1665566 | 2nd   | M      |
|    4 | Sandesh  |  1445777 | 2nd   | M      |
|    3 | Binod    |  1445566 | IV    | M      |
|    1 | Ravi     | 61444333 | first | M      |
|    9 | Pratyush | 61444333 | first | F      |
|    5 | Laxman   | 61444555 | 3rd   | F      |
|    6 | Jerry    | 61444555 | 3rd   | F      |
|   10 | Tom      |   999999 | 2     | M      |
|   10 | Pratyush |   999999 | 2     | M      |
|   10 | Prayash  |   999999 | 2     | M      |
+------+----------+----------+-------+--------+
10 rows in set (0.01 sec)
```

```
mysql> SELECT gender,COUNT(roll)  FROM students GROUP BY gender;
+--------+-------------+
| gender | COUNT(roll) |
+--------+-------------+
| F      |           3 |
| M      |           7 |
+--------+-------------+
2 rows in set (0.00 sec)
```

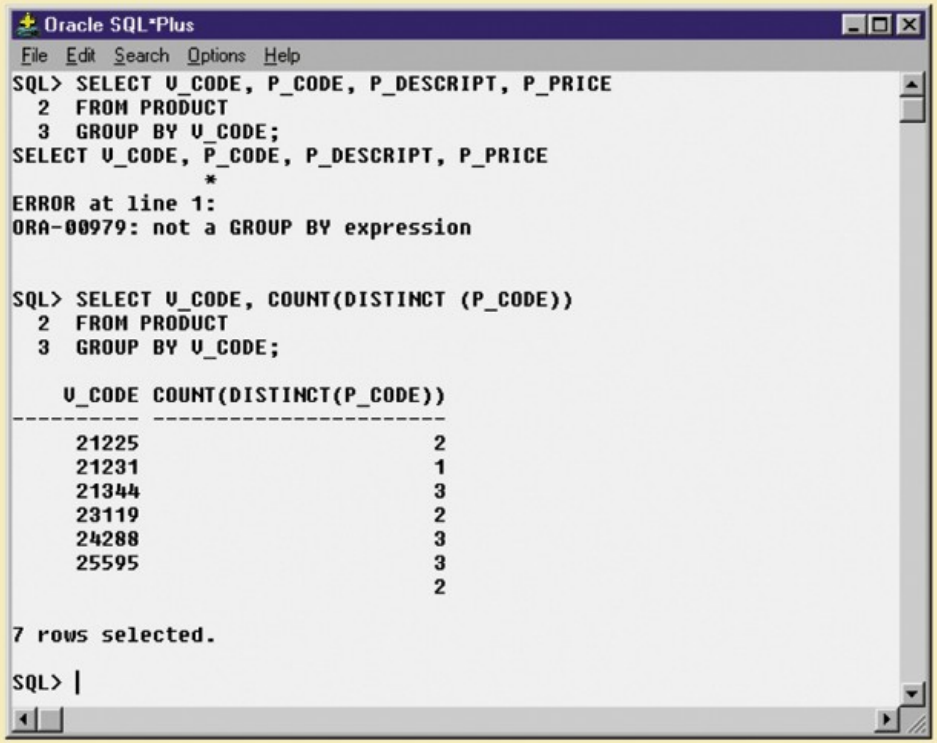
Grouping Data

FIGURE 7.25 GROUP BY clause output examples



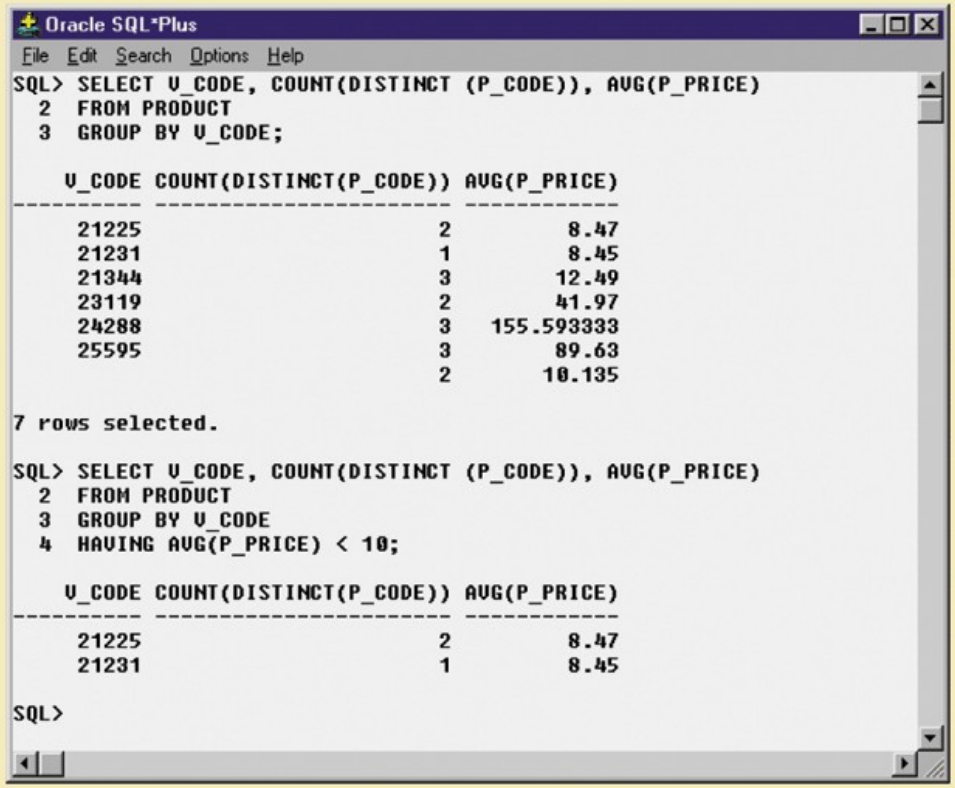
Grouping Data (continued)

FIGURE 7.26 Incorrect and correct use of the GROUP BY clause



Grouping Data (continued)

FIGURE 7.27 An application of the HAVING clause



Virtual Tables: Creating a View

- View is virtual table based on SELECT query
  - Can contain columns, computed columns, aliases, and aggregate functions from one or more tables
- Base tables are tables on which view is based
- Create view by using CREATE VIEW command

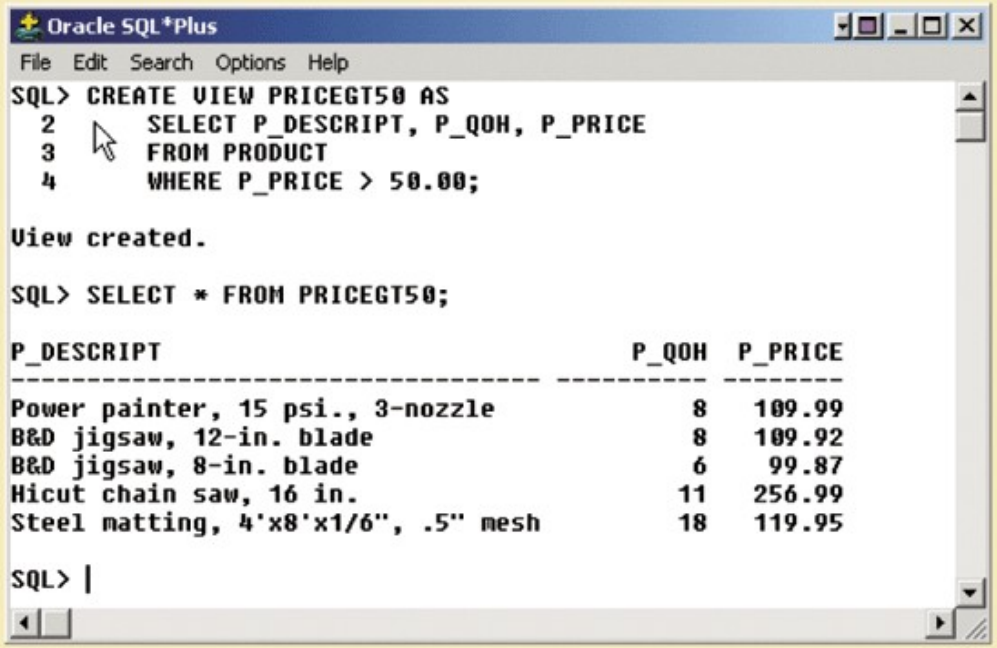
Virtual Tables: Creating a View

```
mysql> create view std_view AS select roll, name from students;
Query OK, 0 rows affected (0.05 sec)
```

```
mysql> select * from std_view;
+------+----------+
| roll | name     |
+------+----------+
|    2 | Manoj    |
|    4 | Sandesh  |
|    3 | Binod    |
|    1 | Ravi     |
|    9 | Pratyush |
|    5 | Laxman   |
|    6 | Jerry    |
|   10 | Tom      |
|   10 | Pratyush |
|   10 | Prayash  |
+------+----------+
10 rows in set (0.00 sec)
```

Virtual Tables: Creating a View (continued)

FIGURE 7.28 Creating a virtual table with the CREATE VIEW command



Joining Database Tables

- Ability to combine (join) tables on common attributes is most important distinction between relational database and other databases
- Join is performed when data are retrieved from more than one table at a time
- Join is generally composed of an equality comparison between foreign key and primary key of related tables

Joining Database Tables (continued)

Cross Join

all rows from all tables

```
mysql> select * from students;
+------+----------+-------+--------+
| roll | name     | sem   | gender |
+------+----------+-------+--------+
|    2 | Manoj    | 2nd   | M      |
|    4 | Sandesh  | 2nd   | M      |
|    3 | Binod    | IV    | M      |
|    1 | Ravi     | first | M      |
|    9 | Pratyush | first | F      |
|    5 | Laxman   | 3rd   | F      |
|    6 | Jerry    | 3rd   | F      |
|   10 | Tom      | 2     | M      |
|   10 | Pratyush | 2     | M      |
|   10 | Prayash  | 2     | M      |
+------+----------+-------+--------+
10 rows in set (0.00 sec)
```

```
mysql> select * from contacts;
+------+---------+---------+----+
| roll | phone   | address | id |
+------+---------+---------+----+
|    1 |  234556 | PKR     |  1 |
|    2 | 1234556 | KTM     |  2 |
|    3 |    6666 | Dharan  |  3 |
+------+---------+---------+----+
3 rows in set (0.00 sec)
```

```
mysql> select s.roll,s.name,c.id,c.phone from students s JOIN  contacts c ;
```

```
mysql> select s.roll,s.name,c.id,c.phone from students s, contacts c;
+------+----------+----+---------+
| roll | name     | id | phone   |
+------+----------+----+---------+
|    2 | Manoj    |  1 |  234556 |
|    2 | Manoj    |  2 | 1234556 |
|    2 | Manoj    |  3 |    6666 |
|    4 | Sandesh  |  1 |  234556 |
|    4 | Sandesh  |  2 | 1234556 |
|    4 | Sandesh  |  3 |    6666 |
|    3 | Binod    |  1 |  234556 |
|    3 | Binod    |  2 | 1234556 |
|    3 | Binod    |  3 |    6666 |
|    1 | Ravi     |  1 |  234556 |
|    1 | Ravi     |  2 | 1234556 |
|    1 | Ravi     |  3 |    6666 |
|    9 | Pratyush |  1 |  234556 |
|    9 | Pratyush |  2 | 1234556 |
|    9 | Pratyush |  3 |    6666 |
|    5 | Laxman   |  1 |  234556 |
|    5 | Laxman   |  2 | 1234556 |
|    5 | Laxman   |  3 |    6666 |
|    6 | Jerry    |  1 |  234556 |
|    6 | Jerry    |  2 | 1234556 |
|    6 | Jerry    |  3 |    6666 |
|   10 | Tom      |  1 |  234556 |
|   10 | Tom      |  2 | 1234556 |
|   10 | Tom      |  3 |    6666 |
|   10 | Pratyush |  1 |  234556 |
|   10 | Pratyush |  2 | 1234556 |
|   10 | Pratyush |  3 |    6666 |
|   10 | Prayash  |  1 |  234556 |
|   10 | Prayash  |  2 | 1234556 |
|   10 | Prayash  |  3 |    6666 |
+------+----------+----+---------+
30 rows in set (0.00 sec)
```

Joining Database Tables (continued)


```
mysql> select * from students;
+------+----------+-------+--------+
| roll | name     | sem   | gender |
+------+----------+-------+--------+
|    2 | Manoj    | 2nd   | M      |
|    4 | Sandesh  | 2nd   | M      |
|    3 | Binod    | IV    | M      |
|    1 | Ravi     | first | M      |
|    9 | Pratyush | first | F      |
|    5 | Laxman   | 3rd   | F      |
|    6 | Jerry    | 3rd   | F      |
|   10 | Tom      | 2     | M      |
|   10 | Pratyush | 2     | M      |
|   10 | Prayash  | 2     | M      |
```

```
mysql> select * from contacts;
+------+---------+---------+----+
| roll | phone   | address | id |
+------+---------+---------+----+
|    1 |  234556 | PKR     |  1 |
|    2 | 1234556 | KTM     |  2 |
|    3 |    6666 | Dharan  |  3 |
+------+---------+---------+----+
3 rows in set (0.00 sec)
```

```
mysql> select s.roll,s.name,c.id,c.phone from students s, contacts c where s.roll=c.roll;
+------+-------+----+---------+
| roll | name  | id | phone   |
+------+-------+----+---------+
|    2 | Manoj |  2 | 1234556 |
|    3 | Binod |  3 |    6666 |
|    3 | Binod | 30 |    6666 |
|    1 | Ravi  |  1 |  234556 |
+------+-------+----+---------+
4 rows in set (0.00 sec)
```

```
mysql> select s.roll,s.name,c.id,c.phone from students s INNER JOIN  contacts c ON  s.roll=c.roll;
+------+-------+----+---------+
| roll | name  | id | phone   |
+------+-------+----+---------+
|    2 | Manoj |  2 | 1234556 |
|    3 | Binod |  3 |    6666 |
|    3 | Binod | 30 |    6666 |
|    1 | Ravi  |  1 |  234556 |
+------+-------+----+---------+
4 rows in set (0.00 sec)
```

Joining Database Tables (continued)


```
mysql> select * from students;
+------+----------+-------+--------+
| roll | name     | sem   | gender |
+------+----------+-------+--------+
|    2 | Manoj    | 2nd   | M      |
|    4 | Sandesh  | 2nd   | M      |
|    3 | Binod    | IV    | M      |
|    1 | Ravi     | first | M      |
|    9 | Pratyush | first | F      |
|    5 | Laxman   | 3rd   | F      |
|    6 | Jerry    | 3rd   | F      |
|   10 | Tom      | 2     | M      |
|   10 | Pratyush | 2     | M      |
|   10 | Prayash  | 2     | M      |
+------+----------+-------+--------+
10 rows in set (0.00 sec)
```

```
mysql> select * from contacts;
+------+---------+---------+----+
| roll | phone   | address | id |
+------+---------+---------+----+
|    1 |  234556 | PKR     |  1 |
|    2 | 1234556 | KTM     |  2 |
|    3 |    6666 | Dharan  |  3 |
+------+---------+---------+----+
3 rows in set (0.00 sec)
```

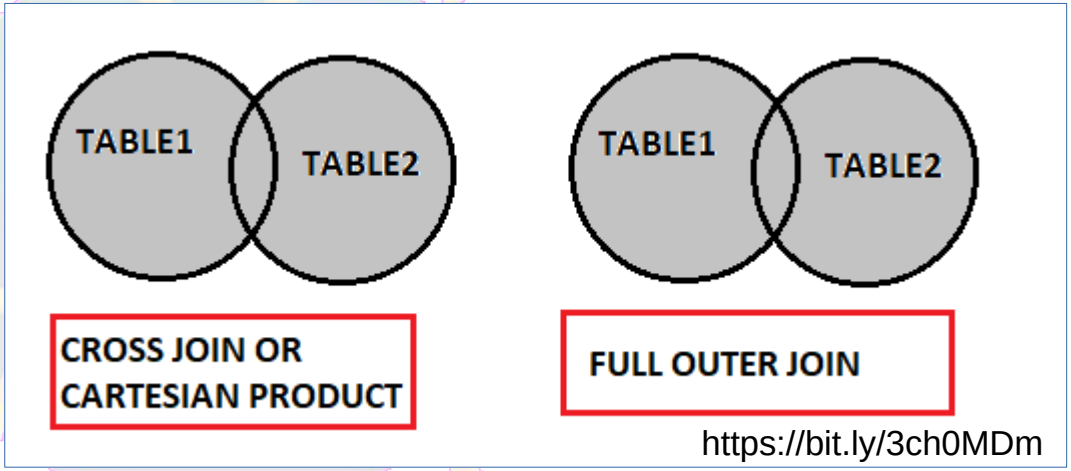
```
mysql> select s.roll,s.name,c.id,c.phone from students s LEFT JOIN  contacts c ON  s.roll=c.roll;
+------+----------+------+---------+
| roll | name     | id   | phone   |
+------+----------+------+---------+
|    1 | Ravi     |    1 |  234556 |
|    2 | Manoj    |    2 | 1234556 |
|    3 | Binod    |    3 |    6666 |
|    3 | Binod    |   30 |    6666 |
|    4 | Sandesh  | NULL |    NULL |
|    9 | Pratyush | NULL |    NULL |
|    5 | Laxman   | NULL |    NULL |
|    6 | Jerry    | NULL |    NULL |
|   10 | Tom      | NULL |    NULL |
|   10 | Pratyush | NULL |    NULL |
|   10 | Prayash  | NULL |    NULL |
+------+----------+------+---------+
11 rows in set (0.00 sec)
```

```
mysql> select s.roll,s.name,c.id,c.phone from students s RIGHT JOIN  contacts c ON  s.roll=c.roll;
```

Joining Database Tables (continued)

Cross Join & Full Outer Join

- A cross join produces a Cartesian product between the two tables,
  - gives all possible combinations of rows.
  - It has no ON clause as we join everything to everything.
- A full outer join is a combination of a left outer and right outer join.
  - It returns all the rows from both tables



Joining Database Tables (continued)

FIGURE 7.29 The results of a join

	P_DESCRIPT	P_PRICE	V_NAME	V_CONTACT	V_AREACODE	V_PHONE
▶	Claw hammer	9.95	Bryson, Inc.	Smithson	615	223-3234
	1.25-in. metal screw, 25	6.99	Bryson, Inc.	Smithson	615	223-3234
	2.5-in. wd. screw, 50	8.45	D&E Supply	Singh	615	228-3245
	7.25-in. pwr. saw blade	14.99	Gomez Bros.	Ortega	615	889-2546
	9.00-in. pwr. saw blade	17.49	Gomez Bros.	Ortega	615	889-2546
	Rat-tail file, 1/8-in. fine	4.99	Gomez Bros.	Ortega	615	889-2546
	Hrd. cloth, 1/4-in., 2x50	39.95	Randsets Ltd.	Anderson	901	678-3998
	Hrd. cloth, 1/2-in., 3x50	43.99	Randsets Ltd.	Anderson	901	678-3998
	B&D jigsaw, 12-in. blade	109.92	ORDVA, Inc.	Hakford	615	898-1234
	B&D jigsaw, 8-in. blade	99.87	ORDVA, Inc.	Hakford	615	898-1234
	Hicut chain saw, 16 in.	256.99	ORDVA, Inc.	Hakford	615	898-1234
	Power painter, 15 psi., 3-nozzle	109.99	Rubicon Syste	Orton	904	456-0092
	B&D cordless drill, 1/2-in.	38.95	Rubicon Syste	Orton	904	456-0092
	Steel matting, 4'x8'x1/6", .5" mesh	119.95	Rubicon Syste	Orton	904	456-0092

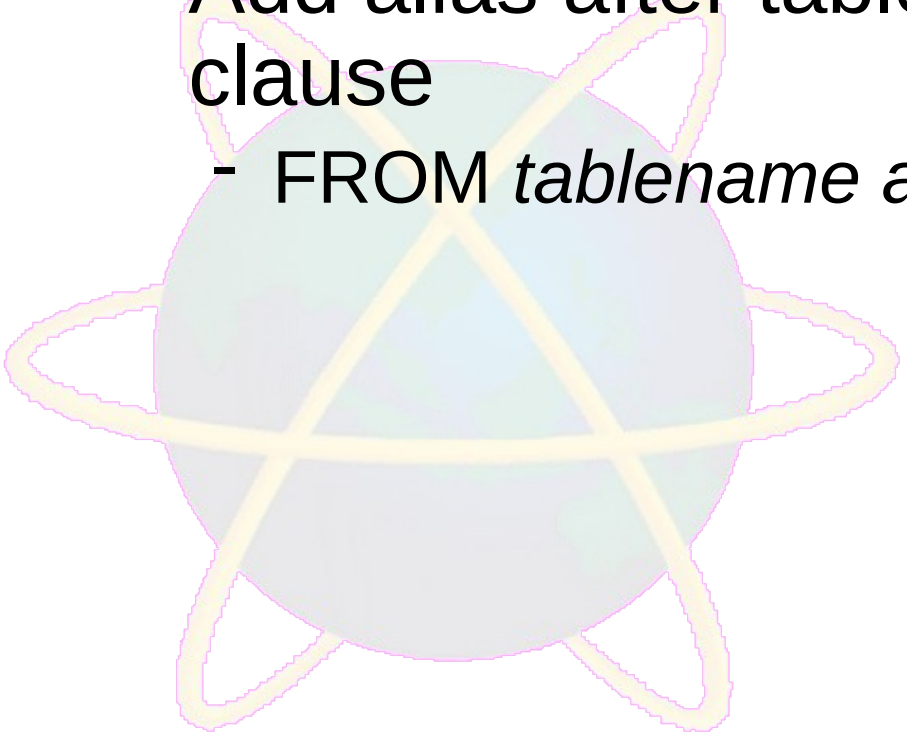
Joining Database Tables (continued)

FIGURE 7.30 An ordered and limited listing after a join

P_DESCRIPT	P_PRICE	V_NAME	V_CONTACT	V_AREACODE	V_PHONE
.25-in. metal screw, 25	6.99	Bryson, Inc.	Smithson	615	223-3234
2.5-in. wd. screw, 50	8.45	D&E Supply	Singh	615	228-3245
Claw hammer	9.95	Bryson, Inc.	Smithson	615	223-3234
B&D cordless drill, 1/2-in.	38.95	Rubicon Systems	Orton	904	456-0092
Steel matting, 4'x8'x1/6", .5" mesh	119.95	Rubicon Systems	Orton	904	456-0092
Hicut chain saw, 16 in.	256.99	ORDVA, Inc.	Hakford	615	898-1234

Joining Tables with an Alias

- Alias can be used to identify source table
- Any legal table name can be used as alias
- Add alias after table name in FROM clause
  - FROM *tablename alias*



Recursive Joins

FIGURE 7.31 The contents of the EMP table

EMP_NUM	EMP_TITLE	EMP_LNAME	EMP_FNAME	EMP_INITIAL	EMP_DOB	EMP_HIRE_DATE	EMP_AREACODE	EMP_PHONE	EMP_MGR
100	Mr.	Kolmycz	George	D	15-Jun-42	15-Mar-85	615	324-5456	
101	Ms.	Lewis	Rhonda	G	19-Mar-65	25-Apr-86	615	324-4472	100
102	Mr.	Vandam	Rhett		14-Nov-58	20-Dec-90	901	675-8993	100
103	Ms.	Jones	Anne	M	16-Oct-74	28-Aug-94	615	898-3456	100
104	Mr.	Lange	John	P	08-Nov-71	20-Oct-94	901	504-4430	105
105	Mr.	Williams	Robert	D	14-Mar-75	08-Nov-98	615	890-3220	
106	Mrs.	Smith	Jeanine	K	12-Feb-68	05-Jan-89	615	324-7883	105
107	Mr.	Diante	Jorge	D	21-Aug-74	02-Jul-94	615	890-4567	105
108	Mr.	Wiesenbach	Paul	R	14-Feb-66	18-Nov-92	615	897-4358	
109	Mr.	Smith	George	K	18-Jun-61	14-Apr-89	901	504-3339	108
110	Mrs.	Genkazi	Leighla	W	19-May-70	01-Dec-90	901	569-0093	108
111	Mr.	Washington	Rupert	E	03-Jan-66	21-Jun-93	615	890-4925	105
112	Mr.	Johnson	Edward	E	14-May-61	01-Dec-83	615	898-4387	100
113	Ms.	Smythe	Melanie	P	15-Sep-70	11-May-99	615	324-9006	105
114	Ms.	Brandon	Marie	G	02-Nov-56	15-Nov-79	901	882-0845	108
115	Mrs.	Saranda	Hermine	R	25-Jul-72	23-Apr-93	615	324-5505	105
116	Mr.	Smith	George	A	08-Nov-65	10-Dec-88	615	890-2984	108

Recursive Joins (continued)

FIGURE 7.32 Using an alias to join a table to itself

	EMP_NUM	A.EMP_LNAME	EMP_MGR	B.EMP_LNAME
▶	112	Johnson	100	Kolmycz
	103	Jones	100	Kolmycz
	102	Vandam	100	Kolmycz
	101	Lewis	100	Kolmycz
	115	Saranda	105	Williams
	113	Smythe	105	Williams
	111	Washington	105	Williams
	107	Diante	105	Williams
	106	Smith	105	Williams
	104	Lange	105	Williams
	116	Smith	108	Wiesenbach
	114	Brandon	108	Wiesenbach
	110	Genkazi	108	Wiesenbach
	109	Smith	108	Wiesenbach

Outer Joins

FIGURE 7.33 The left outer join results

	P_CODE	V_CODE	V_NAME
▶	23109-HB	21225	Bryson, Inc.
	SM-18277	21225	Bryson, Inc.
		21226	SuperLoo, Inc.
	SW-23116	21231	D&E Supply
	13-Q2/P2	21344	Gomez Bros.
	14-Q1/L3	21344	Gomez Bros.
	54778-2T	21344	Gomez Bros.
		22567	Dome Supply
	1546-QQ2	23119	Randsets Ltd.
	1558-QW1	23119	Randsets Ltd.
		24004	Brackman Bros.
	2232/QTY	24288	ORDVA, Inc.
	2232/QWE	24288	ORDVA, Inc.
	89-WRE-Q	24288	ORDVA, Inc.
		25443	B&K, Inc.
		25501	Damal Supplies
	11QER/31	25595	Rubicon Systems
	2238/QPD	25595	Rubicon Systems
	WR3/TT3	25595	Rubicon Systems

Outer Joins (continued)

FIGURE 7.34 The right outer join results

	P_CODE	V_CODE	V_NAME
▶	23114-AA		
	PVC23DRT		
	23109-HB	21225	Bryson, Inc.
	SM-18277	21225	Bryson, Inc.
	SW-23116	21231	D&E Supply
	13-Q2/P2	21344	Gomez Bros.
	14-Q1/L3	21344	Gomez Bros.
	54778-2T	21344	Gomez Bros.
	1546-QQ2	23119	Randsets Ltd.
	1558-QW1	23119	Randsets Ltd.
	2232/QTY	24288	ORDVA, Inc.
	2232/QWE	24288	ORDVA, Inc.
	89-WRE-Q	24288	ORDVA, Inc.
	11QER/31	25595	Rubicon Systems
	2238/QPD	25595	Rubicon Systems
	WR3/TT3	25595	Rubicon Systems

Quick Review Question

- State the difference between types of JOIN below
  - Inner join
  - Left join
  - Right join
  - Full join

Summary of Main Teaching Points

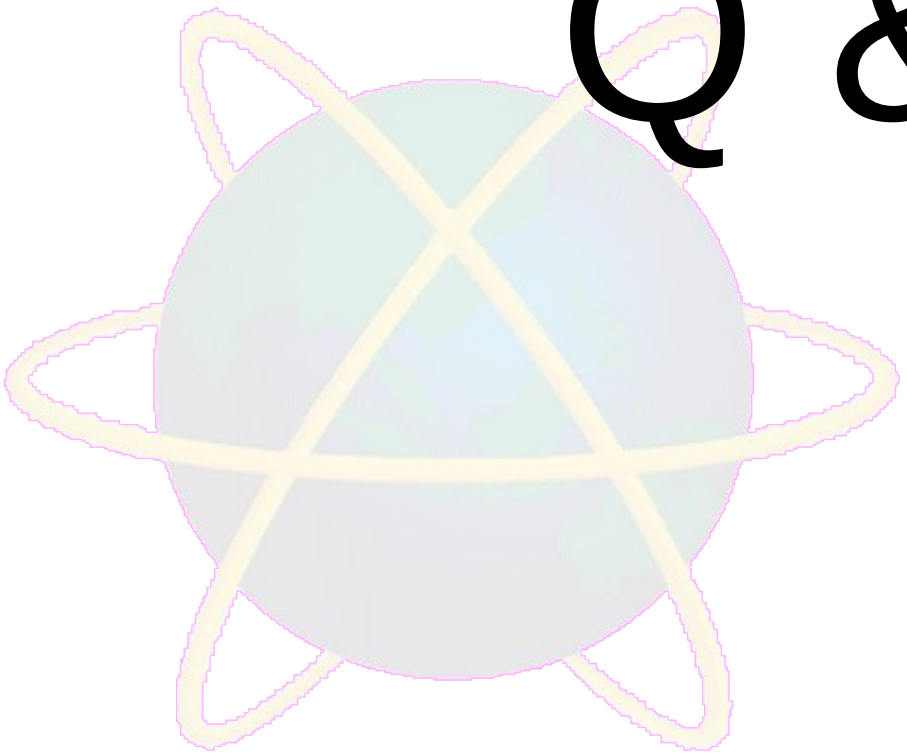
A·P·U
ASIA PACIFIC UNIVERSITY
OF TECHNOLOGY & INNOVATION

- Aggregate functions
  - Special functions that perform arithmetic computations over a set of rows
- ORDER BY clause
  - Used to sort output of SELECT statement
  - Can sort by one or more columns and use either an ascending or descending order
- Join output of multiple tables with SELECT statement
- Natural join uses join condition to match only rows with equal values in specified columns
- Right outer join and left outer join used to select rows that have no matching values in other related table

Question and Answer Session

A·P·U
ASIA PACIFIC UNIVERSITY
OF TECHNOLOGY & INNOVATION

Q & A



What we will cover next

- Degrees of data abstraction
- Extended Entity Relationship Model