

=====

Assignment -1

```

create database dbms;
drop table salespeople;
drop table CUSTOMERS;
use dbms;
create table SALESPEOPLE( Snum int(4),Sname varchar(10),City varchar(10),Comm float(3,2));

insert into SALESPEOPLE values(1001,'Peel','London', 0.12 );
insert into SALESPEOPLE values(1002,'Serres',' San Jose', .13);
insert into SALESPEOPLE values(1004,'Motika','London', .11);
insert into SALESPEOPLE values(1007,'Rifkin','Barcelona', .15);
insert into SALESPEOPLE values(1003,'Axelrod','New York', .10) ;

select * from SALESPEOPLE;
select * from ORDERS;

create table CUSTOMERS(Cnum int(4),Cname varchar(10),City varchar(10),Rating int(4),Snum
int(4));

insert into CUSTOMERS values(2001,'Hoffman','London',100,1001);
insert into CUSTOMERS values(2002,'Giovanni',' Rome', 200,1003);
insert into CUSTOMERS values(2004,'Grass',' Berlin',300, 1002);
insert into CUSTOMERS values(2006,'Clemens','London',100,10011);
insert into CUSTOMERS values(2008,'Cisneros','San Jose',300,1007);
insert into CUSTOMERS values(2007,'Pereira','Rome', 100,1004);
insert into CUSTOMERS values(2003,'Liu San','Jose',200,1002);

create table ORDERS(Onum int(4),Amt float(7,2),Odate date,Cnum int(4),Snum int(4) );
INSERT INTO ORDERS VALUES(3001,18.69,'03-OCT-1990',2008, 1007);

INSERT INTO ORDERS VALUES(3001,18.69,'1990-10-03',2008, 1007);
INSERT INTO ORDERS VALUES(3003,767.19,'1990-10-03',2001,1001);
INSERT INTO ORDERS VALUES(3002,1900.10,'1990-10-03',2007,1004);
INSERT INTO ORDERS VALUES(3005,5160.45,'1990-10-03',2003,1002);
INSERT INTO ORDERS VALUES(3006,1098.16,'1990-10-03',2008,1007);
INSERT INTO ORDERS VALUES(3009,1713.23,'1990-10-04',2002,1003);
INSERT INTO ORDERS VALUES(3007,75.75,'1990-10-04',2004,1002);
INSERT INTO ORDERS VALUES(3008,4723.00,'1990-10-05',2006,1001);
INSERT INTO ORDERS VALUES(3010,309.95,'1990-10-06',2004,1002);
INSERT INTO ORDERS VALUES(3011,9891.88,'1990-10-06',2006,1001);

```

=====

Assignment -2

Introducing Relational Databases.

- 1) Which field of the Customers table is the primary key?
customer number
- 2) What is the 4th column of the Customers table?
Rating
- 3) What is another word for row? For column?
Tuple or record
- 4) Why isn't it possible to see the first five rows of a table?
Because tuples don't have orders.

=====

Assignment -3

Overview of SQL.

1) Does ANSI recognize the data type DATE?

Yes.

2) Which subdivision of SQL is used to insert values in tables?
insert into values

Assignment -4

Retrieving Information from Tables.

1) Write a select command that produces the order number, amount, and date for all rows in the Orders table.

```
select onum, amt, odate from orders;
```

```
mysql> select onum, amt, odate from orders;
```

onum	amt	odate
3001	18.69	1990-10-03
3003	767.19	1990-10-03
3002	1900.10	1990-10-03
3005	5160.45	1990-10-03
3006	1098.16	1990-10-03
3009	1713.23	1990-10-04
3007	75.75	1990-10-04
3008	4723.00	1990-10-05
3010	309.95	1990-10-06
3011	9891.88	1990-10-06

2) Write a query that produces all rows from the Customers table for which the salesperson's number is 1001.

```
mysql> select * from Customers where Snum=1001;
```

Cnum	Cname	City	Rating	Snum
2001	Hoffman	London	100	1001

3) Write a query that displays the Salespeople table with the columns in the following order: city, sname, snum, comm.

```
Select city, sname, snum, comm from salespeople;
```

```
mysql> Select city, sname, snum, comm from salespeople;
```

city	sname	snum	comm
London	Peel	1001	0.12
San Jose	Serres	1002	0.13
London	Motika	1004	0.11
Barcelona	Rifkin	1007	0.15
New York	Axelrod	1003	0.10

4) Write a select command that produces the rating followed by the name of each customer in San Jose.

```
Select rating,cname from customers where city='sanjose';
```

5) Write a query that will produce the snum values of all salespeople from the Orders table (with the duplicate values suppressed).

```
Select distinct snum from salespeople order by snum;
```

snum
1001
1002
1003
1004

```
| 1007 |
+-----+
```

Assignment -5

Relational and Logical Operators.

1) Write a query that will give you all orders for more than Rs. 1,000.

Ans. `select * from orders where amt>1000;`

```
+-----+-----+-----+-----+-----+
| Onum | Amt      | Odate      | Cnum | Snum |
+-----+-----+-----+-----+-----+
| 3002 | 1900.10 | 1990-10-03 | 2007 | 1004 |
| 3005 | 5160.45 | 1990-10-03 | 2003 | 1002 |
| 3006 | 1098.16 | 1990-10-03 | 2008 | 1007 |
| 3009 | 1713.23 | 1990-10-04 | 2002 | 1003 |
| 3008 | 4723.00 | 1990-10-05 | 2006 | 1001 |
| 3011 | 9891.88 | 1990-10-06 | 2006 | 1001 |
+-----+-----+-----+-----+-----+
```

2) Write a query that will give you the names and cities of all salespeople in London with a commission above .10.

Ans. `select sname,city from salespeople where city='London' and comm>0.10;`

```
+-----+-----+
| sname | city |
+-----+-----+
| Peel  | London |
| Motika | London |
+-----+-----+
```

3) Write a query on the Customers table whose output will exclude all customers with a rating <= 100, unless they are located in Rome.

`mysql> Select * from customers where rating>100 or city!='rome';`

```
+-----+-----+-----+-----+-----+
| Cnum | Cname      | City      | Rating | Snum |
+-----+-----+-----+-----+-----+
| 2001 | Hoffman    | London    | 100    | 1001 |
| 2002 | Giovanni   | Rome      | 200    | 1003 |
| 2004 | Grass      | Berlin    | 300    | 1002 |
| 2006 | Clemens    | London    | 100    | 10011 |
| 2008 | Cisneros   | San Jose  | 300    | 1007 |
| 2003 | Liu San    | Jose      | 200    | 1002 |
+-----+-----+-----+-----+-----+
```

4) What will be the output from the following query?

Ans. `Select * from Orders where (amt < 1000 OR NOT (odate = '1990-10-03' AND cnum > 2003));`

```
+-----+-----+-----+-----+-----+
| Onum | Amt      | Odate      | Cnum | Snum |
+-----+-----+-----+-----+-----+
| 3001 | 18.69    | 1990-10-03 | 2008 | 1007 |
| 3003 | 767.19   | 1990-10-03 | 2001 | 1001 |
| 3005 | 5160.45  | 1990-10-03 | 2003 | 1002 |
| 3009 | 1713.23  | 1990-10-04 | 2002 | 1003 |
| 3007 | 75.75    | 1990-10-04 | 2004 | 1002 |
| 3008 | 4723.00  | 1990-10-05 | 2006 | 1001 |
| 3010 | 309.95   | 1990-10-06 | 2004 | 1002 |
| 3011 | 9891.88  | 1990-10-06 | 2006 | 1001 |
+-----+-----+-----+-----+-----+
```

5) What will be the output of the following query?

`Select * from Orders where NOT ((odate = '1990-10-03' OR snum >1006) AND amt >= 1500);`

```
+-----+-----+-----+-----+-----+
| Onum | Amt      | Odate      | Cnum | Snum |
+-----+-----+-----+-----+-----+
| 3001 | 18.69    | 1990-10-03 | 2008 | 1007 |
+-----+-----+-----+-----+-----+
```

3003	767.19	1990-10-03	2001	1001
3006	1098.16	1990-10-03	2008	1007
3009	1713.23	1990-10-04	2002	1003
3007	75.75	1990-10-04	2004	1002
3008	4723.00	1990-10-05	2006	1001
3010	309.95	1990-10-06	2004	1002
3011	9891.88	1990-10-06	2006	1001

6) What **is** a simpler way **to write** this query?

```
Select snum, sname, city, comm from Salespeople Where (comm > .12 or comm < .14);
Select snum, sname, city, comm from Salespeople Where (comm > .12 or comm < .14);
```

snum	sname	city	comm
1001	Peel	London	0.12
1002	Serres	San Jose	0.13
1004	Motika	London	0.11
1007	Rifkin	Barcelona	0.15
1003	Axelrod	New York	0.10

Assignment -6

Using Special Operators in Conditions.

1) Write two different queries that would produce **all** orders taken **on** October 3rd **or** 4th, 1990.

```
mysql> select * from orders where odate='1990-10-03' and '1990-10-03';
```

Onum	Amt	Odate	Cnum	Snum
3001	18.69	1990-10-03	2008	1007
3003	767.19	1990-10-03	2001	1001
3002	1900.10	1990-10-03	2007	1004
3005	5160.45	1990-10-03	2003	1002
3006	1098.16	1990-10-03	2008	1007

2) Write a query that selects **all of** the customers serviced **by** Peel **or** Motika. (Hint:the snum field relates the two tables **to** one another).

```
select * from customers where snum in (select snum from salespeople where sname='Peel' or sname='Motika');
```

Cnum	Cname	City	Rating	Snum
2001	Hoffman	London	100	1001
2007	Pereira	Rome	100	1004

3) Write a query that will produce **all** the customers whose **names begin with** a letter **from** 'A' **to** 'G'.

```
mysql> select * from customers where substr(cname,1,1) between 'A' and 'G';
```

Cnum	Cname	City	Rating	Snum
2002	Giovanni	Rome	200	1003
2004	Grass	Berlin	300	1002
2006	Clemens	London	100	10011
2008	Cisneros	San Jose	300	1007

4 rows in set (0.00 sec)

4) Write a query that selects **all** customers whose **names begin with** the letter 'C'.

```
mysql> select * from customers where cname like 'c%';
```

Cnum	Cname	City	Rating	Snum
------	-------	------	--------	------

2006	Clemens	London	100	10011
2008	Cisneros	San Jose	300	1007

5) Write a query that selects all orders except those with zeroes or NULLs in the amt field.

```
mysql> select * from orders where amt is not null and amt!=0;
```

Onum	Amt	Odate	Cnum	Snum
3001	18.69	1990-10-03	2008	1007
3003	767.19	1990-10-03	2001	1001
3002	1900.10	1990-10-03	2007	1004
3005	5160.45	1990-10-03	2003	1002
3006	1098.16	1990-10-03	2008	1007
3009	1713.23	1990-10-04	2002	1003
3007	75.75	1990-10-04	2004	1002
3008	4723.00	1990-10-05	2006	1001
3010	309.95	1990-10-06	2004	1002
3011	9891.88	1990-10-06	2006	1001

Assignment -7

Summarizing Data with Aggregate Functions.

1) Write a query that counts all orders for October 3.

```
mysql> select count(onum) total_order from orders where odate like '03-OCT-%';
```

total_order
0

2) Write a query that counts the number of different non-NULL city values in the Customers table.

```
mysql> select count(city) from customers where cnum!=null;
```

count(city)
0

3) Write a query that selects each customer's smallest order.

```
mysql> select min(amt) 'smallest order',cnum from orders group by cnum;
```

smallest order	cnum
18.69	2008
767.19	2001
1900.10	2007
5160.45	2003
1713.23	2002
75.75	2004
4723.00	2006

4) Write a query that selects the first customer, in alphabetical order, whose name begins with G.

```
mysql> select * from customers where cname like 'G%' order by cname;
```

Cnum	Cname	City	Rating	Snum
2002	Giovanni	Rome	200	1003
2004	Grass	Berlin	300	1002

5) Write a query that selects the highest rating in each city.

```
mysql> select max(rating) 'highest rating' from customers;
```

```
| highest rating |
+-----+
|          300 |
+-----+
```

6) Write a query that counts the number of salespeople registering orders for each day. (If a salesperson has more than one order on a given day, he or she should be counted only once.).

```
mysql> select count(odate) || 'SALES PERSON REGISTREDDON' "TOTAL", odate from orders group by odate;
```

```
+-----+-----+
| count(odate) || 'SALES PERSON REGISTREDDON' "TOTAL" | odate |
+-----+-----+
|                                                    1 | 1990-10-03 |
|                                                    1 | 1990-10-04 |
|                                                    1 | 1990-10-05 |
|                                                    1 | 1990-10-06 |
+-----+-----+
```

Assignment -8
Formatting Query output.

1) Assume each salesperson has a 12% commission. Write a query on the orders table that will produce the order number, the salesperson number, and the amount of the salesperson's commission for that order.

```
mysql> select onum, snum, amt, (amt*12)/100 commision from orders;
```

```
+-----+-----+-----+-----+
| onum | snum | amt   | commision |
+-----+-----+-----+-----+
| 3001 | 1007 | 18.69 | 2.242800 |
| 3003 | 1001 | 767.19 | 92.062800 |
| 3002 | 1004 | 1900.10 | 228.011997 |
| 3005 | 1002 | 5160.45 | 619.254023 |
| 3006 | 1007 | 1098.16 | 131.779204 |
| 3009 | 1003 | 1713.23 | 205.587598 |
| 3007 | 1002 | 75.75 | 9.090000 |
| 3008 | 1001 | 4723.00 | 566.760000 |
| 3010 | 1002 | 309.95 | 37.194001 |
| 3011 | 1001 | 9891.88 | 1187.025586 |
+-----+-----+-----+-----+
```

2) Write a query on the Customers table that will find the highest rating in each city. Put the output in this form:

For the city (city), the highest rating is : (rating).

```
mysql> select rating, cname, snum from customers order by rating desc;
```

```
+-----+-----+-----+
| rating | cname   | snum |
+-----+-----+-----+
| 300    | Grass   | 1002 |
| 300    | Cisneros | 1007 |
| 200    | Giovanni | 1003 |
| 200    | Liu San  | 1002 |
| 100    | Hoffman  | 1001 |
| 100    | Clemens  | 10011 |
| 100    | Pereira  | 1004 |
+-----+-----+-----+
```

3) Write a query that lists customers in descending order of rating. Output the rating field first, followed by the customer's name and number.

```
mysql> select count(odate) "TOTAL ORDERS", odate from orders group by odate order by "TOTAL ORDERS";
```

```
+-----+-----+
| TOTAL ORDERS | odate |
+-----+-----+
| 5            | 1990-10-03 |
| 2            | 1990-10-04 |
| 1            | 1990-10-05 |
+-----+-----+
```

```
|          2 | 1990-10-06 |  
+-----+-----+
```

4) Write a query that totals the orders for each day and places the results in descending order.

```
=====
```

Assignment - 9

Querying Multiple Tables at Once.

- 1) Write a query that lists each order number followed by the name of the customer who made the order.
- 2) Write a query that gives the names of both the salesperson and the customer for each order along with the order number.
- 3) Write a query that produces all customers serviced by salespeople with a commission above 12%. Output the customer's name, the salesperson's name, and the salesperson's rate of commission.
- 4) Write a query that calculates the amount of the salesperson's commission on each order by a customer with a rating above 100.

```
=====
```

Assignment - 10

Joining a Table to Itself.

- 1) Write a query that produces all pairs of salespeople who are living in the same city. Exclude combinations of salespeople with themselves as well as duplicate rows with the order reversed.
- 2) Write a query that produces the names and cities of all customers with the same rating as Hoffman.

```
=====
```