

Ex. 1:

List the complete PART table.

```
SELECT * FROM PART;
PNUM  PART_DESCRIPTION  UNITS_ON_HAND  ITEM_C  WAR  UNIT_PRICE

AX12   Iron                104            HW      3    21.95
AZ52   Dartboard           20             SG      2    12.95
BA74   Basketball           40             SG      1    29.95
BH22   Cornpopper           95             HW      3    24.95
BT04   Gas Grill            11             AP      2    149.55
BZ66   Washer               52             AP      3    399.99
CA14   Griddle              78             HW      3    39.99
CB03   Bike                 44             SG      1    299.99
CX11   Blender              112            HW      3    22.95
CZ81   Treadmill            68             SG      2    349.95
```

Ex. 2:

List the customer number, last name, first name, and balance for every customer.

```
SELECT CNUM, LAST, FIRST, BALANCE
FROM CUSTOMER;
```

CNUM	LAST	FIRST	BALANCE
124	ADAMS	SALLY	818.75
256	Samuels	Ann	21.5
311	Charles	Don	825.75
315	Daniels	Tom	770.75
405	Williams	Al	402.75
412	Adams	Sally	1817.5
522	Nelson	Mary	98.75
567	Dinh	Tran	402.4
587	Galvez	Mara	114.6
622	Martinz	Dan	1045.75

Ex. 3:

What is the name of customer number 124?

```
SELECT LAST, FIRST  
FROM CUSTOMER  
WHERE CNUM = '124';
```

```
LAST    FIRST  
ADAMS   SALLY
```

Ex.4:

Give the order number for every order placed by customer number 124 on 9/05/98

```
SELECT * FROM ORDERS
WHERE CNUM = '124'
AND ORDER_DATE = '05-SEP-98';
```

ONUM	ORDER_DATE	CNUM
12500	05-SEP-98	124

Ex.5: List the part number and part description for every part that is not item class

7

HW.

```
SELECT PNUM, PART_DESCRIPTION
FROM PART
WHERE ITEM_CLASS NOT IN ('HW');
```

PNUM	PART_DESCRIPTION
AZ52	Dartboard
BA74	Basketball
BT04	Gas Grill
BZ66	Washer
CB03	Bike
CZ81	Treadmill

Ex.6:

List the customer number, last name, first name, and balance for every customer whose balance is between 500 and 1000. (Use two ways)

**SELECT CNUM, LAST, FIRST, BALANCE
FROM CUSTOMER
WHERE BALANCE BETWEEN 500 AND 1000;**

CNUM	LAST	FIRST	BALANCE
124	ADAMS	SALLY	818.75
311	Charles	Don	825.75
315	Daniels	Tom	770.75

**SELECT CNUM, LAST, FIRST, BALANCE
FROM CUSTOMER
WHERE BALANCE > 500
AND BALANCE < 1000;**

CNUM	LAST	FIRST	BALANCE
124	ADAMS	SALLY	818.75
311	Charles	Don	825.75
315	Daniels	Tom	770.75

Ex.7:

Give the part number , part description, on-hand value (units on hand * unit Price)

```
SELECT PNUM, PART_DESCRIPTION, (UNITS_ON_HAND * UNIT_PRICE) AS ON_HAND_VALUE  
FROM PART;
```

PNUM	PART_DESCRIPTION	ON_HAND_VALUE
AX12	Iron	2282.8
AZ52	Dartboard	259
BA74	Basketball	1198
BH22	Cornpopper	2370.25
BT04	Gas Grill	1645.05
BZ66	Washer	20799.48
CA14	Griddle	3119.22
CB03	Bike	13199.56
CX11	Blender	2570.4
CZ81	Treadmill	23796.6

Ex.8:

List the part number and part description for every part whose item class is HW or SG. (Use in operator)

```
SELECT PNUM, PART_DESCRIPTION
FROM PART
WHERE ITEM_CLASS = 'HW'
OR ITEM_CLASS = 'SG';
PNUM    PART_DESCRIPTION
AX12    Iron
AZ52    Dartboard
BA74    Basketball
BH22    Cornpopper
CA14    Griddle
CB03    Bike
CX11    Blender
CZ81    Treadmill
```

Ex. 9:

Find the customer number, last name, and first name for every customer whose first name begins with letter “D”.

```
SELECT CNUM, LAST, FIRST
FROM CUSTOMER
WHERE FIRST LIKE '%D_____%';
```

CNUM	LAST	FIRST
311	Charles	Don
622	Martinz	Dan

Ex. 10:

List all detail about all parts. Order the output by part description.

```
SELECT PNUM, PART_DESCRIPTION, UNITS_ON_HAND, ITEM_CLASS, WAREHOUSE_NUMBER, UNIT_PRICE
FROM PART
ORDER BY PART_DESCRIPTION;
```

PNUM	PART_DESCRIPTION	UNITS_ON_HAND	ITEM_C	WAR	UNIT_PRICE
BA74	Basketball	40	SG	1	29.95
CB03	Bike	44	SG	1	299.99
CX11	Blender	112	HW	3	22.95
BH22	Cornpopper	95	HW	3	24.95
AZ52	Dartboard	20	SG	2	12.95
BT04	Gas Grill	11	AP	2	149.55
CA14	Griddle	78	HW	3	39.99
AX12	Iron	104	HW	3	21.95
CZ81	Treadmill	68	SG	2	349.95

Ex. 11:

Find out how many customers have a balance that is less than their credit limit.

```
SELECT COUNT (*)  
FROM CUSTOMER  
WHERE BALANCE < CREDIT_LIMIT;
```

```
COUNT(*)  
8
```

Ex. 13:

Find the number of customers and the total of their balances.

```
SELECT COUNT (*), SUM(BALANCE)  
FROM CUSTOMER;
```

```
COUNT(*)    SUM(BALANCE)  
10          6318.5
```


Ex. 14:

Find the customer number of every customer who currently has an open order.

```
SELECT DISTINCT CNUM  
FROM ORDERS  
ORDER BY CNUM;
```

```
CNUM  
124  
256  
311  
315  
522
```

```
SELECT DISTINCT CUSTOMER.CNUM, CUSTOMER.LAST, CUSTOMER.FIRST, ORDERS.ONUM  
FROM CUSTOMER  
INNER JOIN ORDERS  
ON CUSTOMER.CNUM=ORDERS.CNUM  
ORDER BY CUSTOMER.LAST;
```

CNUM	LAST	FIRST	ONUM
124	ADAMS	SALLY	12500
124	ADAMS	SALLY	12489
311	Charles	Don	12491
315	Daniels	Tom	12494
522	Nelson	Mary	12498
522	Nelson	Mary	12504
256	Samuels	Ann	12495

Ex. 15:

List the part number, part description, and units on hand of all parts whose number of units on hand is more than the average.

```
SELECT PNUM, PART_DESCRIPTION, UNITS_ON_HAND
FROM PART
WHERE UNITS_ON_HAND >
(SELECT AVG(UNITS_ON_HAND)
FROM PART);
```

PNUM	PART_DESCRIPTION	UNITS_ON_HAND
AX12	Iron	104
BH22	Cornpopper	95
CA14	Griddle	78
CX11	Blender	112
CZ81	Treadmill	68

Ex. 16:

List the total for each order.

```
SELECT ONUM,
SUM(NUMBER_ORDERED*QUOTED_PRICE)
FROM ORDER_LINE
GROUP BY ONUM
ORDER BY ONUM;
```

ONUM	SUM(NUMBER_ORDERED*QUOTED_PRICE)
12489	241.45
12491	549.98
12494	1119.96

12495	45.9
12498	125.7
12500	149.99
12504	651.98

Ex. 17:

List the total of those orders over 200.

```
SELECT ONUM,  
SUM(NUMBER_ORDERED*QUOTED_PRICE)  
FROM ORDER_LINE  
GROUP BY ONUM  
HAVING SUM(NUMBER_ORDERED*QUOTED_PRICE) > 200;
```

ONUM	SUM(NUMBER_ORDERED*QUOTED_PRICE)
12494	1119.96
12491	549.98
12489	241.45
12504	651.98

Ex. 18:

List the customer number, last name, and first name for every customer whose street value is null (unknown).

First had to update table to include "null" street values.

Eg.:

```
UPDATE CUSTOMER  
SET STREET = NULL  
WHERE CNUM = '311';
```

```
SELECT CNUM, LAST, FIRST  
FROM CUSTOMER  
WHERE STREET IS NULL;
```

CNUM	LAST	FIRST
124	Adams	Sally
311	Charles	Don

5.2 Multiple-Table Queries

Join two tables.

Ex. 19:

List the customer number, last name, and first name for every customer together with the sales rep number, last name, and first name for the sales rep who represents each customer.

```
SELECT CNUM, CUSTOMER.LAST, CUSTOMER.FIRST,  
SALES_REP.SNUM, SALES_REP.LAST, SALES_REP.FIRST  
FROM CUSTOMER, SALES_REP  
WHERE CUSTOMER.SNUM = SALES_REP.SNUM;
```

CNUM	LAST	FIRST	SNUM	LAST	FIRST
124	Adams	Sally	03	Jones	Mary
256	Samuels	Ann	06	Smith	William
311	Charles	Don	12	Diaz	Miguel
315	Daniels	Tom	06	Smith	William
405	Williams	Al	12	Diaz	Miguel
412	Adams	Sally	03	Jones	Mary
522	Nelson	Mary	12	Diaz	Miguel
567	Dinh	Tran	06	Smith	William
587	Galvez	Mara	06	Smith	William
622	Martinz	Dan	03	Jones	Mary

10 rows selected.

Using IN operator.

Ex. 20:

Find the customer number of every customer who currently has an open order.

```
SELECT DISTINCT CNUM  
FROM ORDERS;
```

CNUM

311

256

124

315

522

OR:

```
SELECT CNUM, LAST, FIRST  
FROM CUSTOMER  
WHERE CNUM IN  
(SELECT CNUM FROM ORDERS);
```

CNUM	LAST	FIRST
124	Adams	Sally
256	Samuels	Ann
311	Charles	Don
315	Daniels	Tom
522	Nelson	Mary

Using EXISTS operator.

Ex. 20:

Find the order number and order date for every order that contains part number BT04

```
SELECT ONUM, ORDER_DATE
FROM ORDERS
WHERE EXISTS
(SELECT *
FROM ORDER_LINE
WHERE ORDERS.ONUM =
ORDER_LINE.ONUM
AND PNUM = 'BT04');
```

ONUM	ORDER_DATE
12491	02-SEP-98
12500	05-SEP-98

Joining a Table to Itself

Ex. 21:

Find every pair of customers who have the *SAME* first and last name.

```
SELECT F.CNUM, F.LAST, F.FIRST,
S.CNUM, S.LAST, S.FIRST
FROM CUSTOMER F, CUSTOMER S
WHERE F.LAST = S.LAST
```

```
AND F.FIRST = S.FIRST
AND F.CNUM < S.CNUM;
```

```
CNUM   LAST   FIRST   CNUM   LAST   FIRST
124 Adams  Sally   412 Adams  Sally
```

Set Operations

Ex. 22:

List the customer number for every customer who is either represented by sales rep number 12 or who currently has orders on file.

```
SELECT CNUM, LAST, FIRST
FROM CUSTOMER
WHERE SNUM = '12'
UNION
SELECT CUSTOMER.CNUM, LAST, FIRST
FROM CUSTOMER, ORDERS
WHERE CUSTOMER.CNUM =
ORDERS.CNUM;
```

```
CNUM   LAST   FIRST
124    Adams  Sally
256    Samuels Ann
311    Charles Don
315    Daniels Tom
405    Williams Al
522    Nelson  Mary
```


Ex. 23:

List the customer number for every customer who is either represented by sales rep number 12 and who currently has orders on file.

```
SELECT CNUM, LAST, FIRST
FROM CUSTOMER
WHERE SNUM = '12'
INTERSECT
SELECT CUSTOMER.CNUM, LAST, FIRST
FROM CUSTOMER, ORDERS
WHERE CUSTOMER.CNUM =
ORDERS.CNUM;
```

CNUM	LAST	FIRST
311	Charles	Don
522	Nelson	Mary

Ex. 24:

List the customer number for every customer who is either represented by sales rep number 12 and who does not have orders currently on file.

```
SELECT CNUM, LAST, FIRST
FROM CUSTOMER
WHERE SNUM = '12'
MINUS
SELECT CUSTOMER.CNUM, LAST, FIRST
FROM CUSTOMER, ORDERS
```

```
WHERE CUSTOMER.CNUM =  
ORDERS.CNUM;
```

CNUM	LAST	FIRST
405	Williams	Al

ALL and ANY

Ex. 25:

Find the customer number, last name, and first name for every customer whose balance is larger than the individual balances of every customer of sales rep 12

```
SELECT CNUM, LAST, FIRST  
FROM CUSTOMER  
WHERE BALANCE > ALL  
(SELECT BALANCE  
FROM CUSTOMER  
WHERE SNUM = '12');
```

CNUM	LAST	FIRST
412	Adams	Sally
622	Martinz	Dan

Ex. 26:

Find the customer number, last name, and first name for every customer whose balance is larger than the balance of at least one customer of sales rep 12

```
SELECT CNUM, LAST, FIRST
```

```
FROM CUSTOMER
WHERE BALANCE > ANY
(SELECT BALANCE
FROM CUSTOMER
WHERE SNUM = '12');
```

CNUM	LAST	FIRST
412	Adams	Sally
622	Martinz	Dan
311	Charles	Don
124	Adams	Sally
315	Daniels	Tom
405	Williams	Al
567	Dinh Tran	
587	Galvez	Mara

Ex. 27:

Change the last name of customer number 256 to Jones

```
UPDATE CUSTOMER
SET LAST = 'Jones'
WHERE CNUM = '256';
1 row updated.
SELECT CNUM, LAST, FIRST
FROM CUSTOMER
WHERE CNUM = '256';
```

CNUM	LAST	FIRST
256	Jones	Ann

Ex. 28:

Delete from the database the customer information for 'Williams'

```
DELETE CUSTOMER  
WHERE LAST = 'Williams';
```

1 row deleted.

Creating a New Table from an Existing Table**Ex. 29:**

Create a new table where this table has Customer2 as the name and has the same structure and data as customer table

```
CREATE TABLE CUSTOMER2  
(CNUM CHAR(3) PRIMARY KEY,  
LAST CHAR(10) NOT NULL,  
FIRST CHAR(8) NOT NULL,  
STREET CHAR(15),  
CITY CHAR(15),  
STATE CHAR(2),  
ZIP_CODE CHAR(5),  
BALANCE DECIMAL(7,2),  
CREDIT_LIMIT DECIMAL(6,2),  
SNUM CHAR(2) CONSTRAINT C10 REFERENCES SALES_REP(SNUM));
```

```
INSERT INTO CUSTOMER2  
SELECT * FROM CUSTOMER;
```

9 rows created.

CNUM	LAST	FIRST	STREET	CITY	STATE	ZIP_CODE	BALANCE	CREDIT_LIMIT	SNUM
124	Adams	Sally		Lansing	MI	49224	818.75	1000	03
256	Jones	Ann	215 Pete	Grant	MI	49219	21.5	1500	06
311	Charles	Don		Ira	MI	49034	825.75	1000	12
315	Daniels	Tom	914 Cherry	Kent	MI	48391	770.75	750	06
412	Adams	Sally	16 Elm	Lansing	MI	49224	1817.5	2000	03
522	Nelson	Mary	108 Pine	Ada	MI	49441	98.75	1500	12
567	Dinh	Tran	808 Ridge	Harper	MI	48421	402.4	750	06
587	Galvez	Mara	512 Pine	Ada	MI	49441	114.6	1000	06
622	Martinz	Dan	419 Chip	Grant	MI	49219	1045.75	1000	03

Ex. 30:

Assume that we have an already created new table called **Small_Customers** that has the same structure as **Customer** table. How can you insert in the new table only those rows which the credit limit is 1200 or less.

```
INSERT INTO SMALL_CUSTOMER
SELECT *
FROM CUSTOMER
WHERE CREDIT_LIMIT <= 1200;
```

Ex. 31:

Modify the table customer by adding a new column called customer_type where this field will be used to classify customers as Regular, Distributor, or Special customers. We will use 'R', 'D', or 'S' letter to represent each customer.

```
ALTER TABLE CUSTOMER  
ADD CUSTOMER_TYPE CHAR(1);
```

Ex. 32:

Modify the table customer by changing the length of street field from 15 characters to 20.

```
ALTER TABLE CUSTOMER  
MODIFY STREET CHAR(20);
```

Table altered.

Ex. 33:

Modify the table customer by deleting the ZIP_CODE column.

```
ALTER TABLE CUSTOMER  
DROP COLUMN ZIP_CODE;
```

Could not make this syntax work:

```
ALTER TABLE CUSTOMER  
DELETE COLUMN ZIP_CODE;
```

Database Administration

Ex. 34:

Define a view named HOUSEWARES that consists of the part number, part description, units on hand, and unit price of all parts in item class HW.

```
CREATE VIEW HOUSEWARES AS
SELECT PNUM, PART_DESCRIPTION, UNITS_ON_HAND, UNIT_PRICE
FROM PART
WHERE ITEM_CLASS = 'HW';
```

PNUM	PART_DESCRIPTION	UNITS_ON_HAND	UNIT_PRICE
AX12	Iron	104	21.95
BH22	Cornpopper	95	24.95
CA14	Griddle	78	39.99
CX11	Blender	112	22.95

Ex. 35:

Define a view named HOUSEWARES that consists of the part number, part description, units on hand, and unit price of all parts in item class HW. In this view, rename the PNUM column to NUM, the PART_DESCRIPTION column to DSC, the UNITS_ON_HAND to OH, and UNIT_PRICE to PRCE

```
CREATE VIEW HOUSEWARES (NUM, DSC, OH, PRCE) AS
SELECT PNUM, PART_DESCRIPTION, UNITS_ON_HAND, UNIT_PRICE
FROM PART
WHERE ITEM_CLASS = 'HW';
```

NUM	DSC	OH	PRCE
AX12	Iron	104	21.95
BH22	Cornpopper	95	24.95
CA14	Griddle	78	39.99
CX11	Blender	112	22.95

Ex. 36:

Drop the view HOUSEWARES

```
DROP VIEW HOUSEWARES;
```

View dropped.

Security

Assume that we have four users in our database, Smith, Jones, Brown and your user.

To give a user(s) a specific right(s) on a database table(s), use the grant command.

Syntax:

```
Grant [ Right ] on [table(s)] to [user(s) ]
```

Right could be one of the following: Select, Insert, Update, Delete, Index, Alter or ALL

Ex. 36:

Give users Smith and Brown right to add new parts to the PART table

```
GRANT INSERT  
ON PART  
TO SMITH, BROWN;
```

Ex. 37:

Give user Jones all rights on CUSTOMER table

```
GRANT ALL  
ON CUSTOMER  
TO JONES;
```

Ex. 38:

Give user Smith right to change the last name, first name of customers

```
GRANT UPDATE (LAST, FIRST)  
ON CUSTOMER  
TO SMITH;
```

To take a specific right(s) on a database table(s) from a user(s), use the revoke command.

Syntax:

```
Revoke [Right] on [table(s)] from [user(s)]
```

Ex. 39:

Prevent user Jones from retrieving data form Customer table

```
REVOKE SELECT ON CUSTOMER FROM JONES;
```

Indexes

Index is used to speed up the searching process significantly.

Syntax: CREATE INDEX INDEX_NAME ON TABLE_NAME (FIELD(S))

Ex. 40:

Create an index named CUSTNAME on the combination of the LAST and FIRST

```
CREATE INDEX CUSTNAME ON CUSTOMER(LAST, FIRST);
```

Index created.

To drop an index, use

Syntax: DROP INDEX INDEX_NAME

Using the word **unique** in the create index command will prevent having repetition in the index values. This is called unique index.

The System Catalog

Ex. 41

List the name of every table for which the owner is Smith

```
SELECT *  
FROM DBA_TABLES  
WHERE OWNER = 'Smith';
```

Ex. 42

List the name of every view for which the owner is 'Jones'

```
SELECT *  
FROM DBA_VIEWS  
WHERE OWNER = 'Jones';
```

Ex. 43

List every table owned by 'Jones' that contains a column named CNUM

```
SELECT *  
FROM DBA_TABLES  
WHERE OWNER = 'Jones'  
AND TABLE_NAME = 'CNUM';
```