DBMS LAB

P.SONY PRIYA

NO.19PA1A05E9

CSE-C

SQL Commands:

• The standard SQL commands to interact with relational databases are CREATE, SELECT, INSERT, UPDATE, DELETE and DROP.

• These commands can be classified into groups based on their nature:

DDL - Data Definition Language:

• DDL defines the conceptual schema providing a link between the logical and the physical structure of the database.

• Few of the basic commands for DDL are:

Command	Description
CREATE	Creates a new table, a view of a table, or other object in database
DROP	Deletes an entire table, a view of a table or other object in the database.
TRUNCATE	Deletes the data inside a table, but not the table itself.
ALTER	Modifies an existing database object, such as a table.

DML - Data Manipulation Language:

• DML provides the data manipulation techniques like selection, insertion, deletion, updation, modification, replacement, retrieval, sorting and display of data or records.

• Few of the basic commands for DML are:

Command	Description
INSERT	Creates a record
SELECT	Retrieves certain records from one or more tables
UPDATE	Modifies records
DELETE	Deletes records

DCL - Data Control Language:

• These SQL commands are used for providing security to database objects.

• The different DCL commands are:

Command	Description
GRANT	Gives a privilege to user
REVOKE	Takes back privileges granted from user

TCL – Transaction Control Language:

• It includes commands to control the transactions in a database system.

• The commonly used commands are:

Command	Description
COMMIT	Make all the changes made by the statements issued permanent.
ROLLBACK	Undoes all changes since the beginning of transaction or since a save point.

Implementation of SQL Commands

CREATE TABLE

```
CREATE TABLE EMPLOYEE(
                  EMID NUMBER,
                  ENAME VARCHAR2(30),
                  AGE NUMBER,
                  SALARY NUMBER,
                  DEPT VARCHAR2(30)
                  )
```

Table created.

DESCRIBE

```
DESCRIBE EMPLOYEE
```

TABLE EMPLOYEE

Column	Null?	Type
EMID	-	NUMBER
ENAME	-	VARCHAR2(30)
AGE	-	NUMBER
SALARY	-	NUMBER
DEPT	-	VARCHAR2(30)

5 rows selected.

ALTER Statement:

```
ALTER TABLE EMP ADD (ADDRESS VARCHAR2(30))
```

Table altered.

```
ALTER TABLE EMP MODIFY (ADDRESS VARCHAR2(40))
```

Table altered.

```
ALTER TABLE EMP DROP (ADDRESS)
```

Table altered.

DROP TABLE:

```
DROP TABLE EMPLOYEE
```
Table dropped.

INSERT:

```
INSERT INTO EMPLOYEE VALUES(1,'JANU',25,25000,'IT')
```
1 row(s) inserted.

```
INSERT INTO EMPLOYEE VALUES(2,'RAMU',28,40000,'CSE')
```
1 row(s) inserted.

```
INSERT INTO EMPLOYEE VALUES(3,'AJAY',26,55000,'ECE')
```
1 row(s) inserted.

```
INSERT INTO EMPLOYEE VALUES(4,'HARI',27,34000,'IT')
```
1 row(s) inserted.

```
INSERT INTO EMPLOYEE VALUES(5,'SAKI',26,43000,'CSE')
```
1 row(s) inserted.

```
INSERT INTO EMPLOYEE VALUES(6,'VIJAY',25,25000,'IT')
```
1 row(s) inserted.

```
INSERT INTO EMPLOYEE VALUES(7,'DEEPU',27,55000,'ECE')
```
1 row(s) inserted.

```
SELECT * FROM EMPLOYEE
```

EMID	ENAME	AGE	SALARY	DEPT
4	HARI	27	34000	IT
5	SAKI	26	43000	CSE
1	JANU	25	25000	IT
3	AJAY	26	55000	ECE
2	RAMU	28	40000	CSE
6	VIJAY	25	25000	IT
7	DEEPU	27	55000	ECE

7 rows selected.

UPDATE:

```
UPDATE EMPLOYEE SET AGE=24 WHERE EMID=1
```
1 row(s) updated.
```
UPDATE EMPLOYEE SET ENAME='AKASH' WHERE EMID=7
```
1 row(s) updated.

DELETE command:

```
DELETE EMPLOYEE WHERE EMID=4
```
1 row(s) deleted.

SELECT:

```
SELECT * FROM EMPLOYEE
```

EMID	ENAME	AGE	SALARY	DEPT
5	SAKI	26	43000	CSE
1	JANU	24	25000	IT
3	AJAY	26	55000	ECE
2	RAMU	28	40000	CSE
6	VIJAY	25	25000	IT
7	AKASH	27	55000	ECE

6 rows selected.

DISTINCT:

```
SELECT DISTINCT DEPT FROM EMPLOYEE ORDER BY DEPT
```

DEPT
CSE
ECE
IT

3 rows selected.

WHERE clause – (Extracting specific rows)

```
SELECT ENAME, AGE,DEPT FROM EMPLOYEE WHERE SALARY>40000
```

ENAME	AGE	DEPT
SAKI	26	CSE
AJAY	26	ECE
AKASH	27	ECE

3 rows selected.

The AND Operator:

```
SELECT ENAME, AGE,EMID FROM EMPLOYEE WHERE SALARY>40000 AND DEPT='ECE'
```

ENAME	AGE	EMID
AJAY	26	3
AKASH	27	7

2 rows selected.

The OR Operator:

```
SELECT ENAME, AGE,EMID FROM EMPLOYEE WHERE SALARY>40000 OR DEPT='ECE'
```

ENAME	AGE	EMID
SAKI	26	5
AJAY	26	3
AKASH	27	7

3 rows selected.

ORDER BY – (Sorting the data)

Ascending order

```
SELECT * FROM EMPLOYEE ORDER BY ENAME
```

EMID	ENAME	AGE	SALARY	DEPT

3	AJAY	26	55000	ECE
7	AKASH	27	55000	ECE
1	JANU	24	25000	IT
2	RAMU	28	40000	CSE
5	SAKI	26	43000	CSE
6	VIJAY	25	25000	IT

6 rows selected.

Descending order

```
SELECT * FROM EMPLOYEE ORDER BY ENAME DESC
```

EMID	ENAME	AGE	SALARY	DEPT
6	VIJAY	25	25000	IT
5	SAKI	26	43000	CSE
2	RAMU	28	40000	CSE
1	JANU	25	25000	IT
4	HARI	27	34000	IT
7	DEEPU	27	55000	ECE
3	AJAY	26	55000	ECE

7 rows selected.

TRUNCATE COMMAND:

```
TRUNCATE TABLE EMPLOYEE
```

Table truncated

```
DESC EMPLOYEE
```

TABLE EMPLOYEE

Column	Null?	Type
EMID	-	NUMBER
ENAME	-	VARCHAR2(30)
AGE	-	NUMBER

SALARY - NUMBER

DEPT - VARCHAR2(30)

5 rows selected.

SQL Functions:

• The SQL functions serve the purpose of manipulating data items and returning a result.
• There are many built in functions included in SQL and can be classified as Group Functions and
Scalar Functions.
• **Group Functions**: **o** Functions that act on set of values are called group functions.
o A group functions can takes entire column of data as its arguments and produces a single data item that summarizes the column.
o Following are the SQL group functions.

Function	Description
AVG	Returns average value of,,N", ignoring NULL values
COUNT(expr)	Returns the number of rows where ,,expr" is not NULL
COUNT(*)	Returns the number of rows in the table including duplicates and those with NULL values
MIN	Returns minimum value of ,,expr"
MAX	Returns maximum value of ,,expr"
SUM	Returns sum of values ,,N"

```
CREATE TABLE EMPLOYEE(
                    ID NUMBER,
                    ENAME VARCHAR(30),
                    AGE NUMBER,
                    JOB VARCHAR(30),
                    CITY VARCHAR(30),
                    SALARY NUMBER
                    )
```

Table created.

```
INSERT INTO EMPLOYEE VALUES(7369,'SMITH',30,'CLECK','PUNE',25000)
```

1 row(s) inserted.

```
INSERT INTO EMPLOYEE VALUES(7587,'ALIAN',29,'MANAGER','MUMBAI',35000)
```
1 row(s) inserted.

```
INSERT INTO EMPLOYEE VALUES(7645,'WALD',28,'MANAGER','MUMBAI',35000)
```
1 row(s) inserted.

```
INSERT INTO EMPLOYEE
VALUES(7854,'FORD',25,'ANALYST','HYDERABAD',65000)
```
1 row(s) inserted.

```
INSERT INTO EMPLOYEE VALUES(7456,'TOMMY',32,'CLERK','PUNE',26000)
```
1 row(s) inserted.

```
INSERT INTO EMPLOYEE
VALUES(7895,'ADAMS',29,'ANALYST','HYDERABAD',55000)
```
1 row(s) inserted.

```
INSERT INTO EMPLOYEE
VALUES(7774,'HARRY',25,'SALESMAN','BANGLORE',25000)
```
1 row(s) inserted.

```
INSERT INTO EMPLOYEE
VALUES(7344,'HARI',27,'SALESMAN','BANGLORE',25000)
```
1 row(s) inserted.

```
SELECT * FROM EMPLOYEE
```

ID	ENAME	AGE	JOB	CITY	SALARY
7369	SMITH	30	CLECK	PUNE	25000
7645	WALD	28	MANAGER	MUMBAI	35000
7854	FORD	25	ANALYST	HYDERABAD	65000
7895	ADAMS	29	ANALYST	HYDERABAD	55000
7587	ALIAN	29	MANAGER	MUMBAI	35000
7344	HARI	27	SALESMAN	BANGLORE	25000
7456	TOMMY	32	CLERK	PUNE	26000
7774	HARRY	25	SALESMAN	BANGLORE	25000

8 rows selected.

AVG ( ) Function:

```
SELECT AVG(SALARY) FROM EMPLOYEE
```

AVG(SALARY)

36375

```
SELECT AVG(SALARY) FROM EMPLOYEE WHERE CITY='PUNE'
```

AVG(SALARY)

25500

MAX ( ) Function:

```
SELECT MAX(AGE) FROM EMPLOYEE
```

MAX(AGE)

32

MIN ( ) Function:

```
SELECT MIN(AGE) FROM EMPLOYEE
```

MIN(AGE)

25

SUM ( ) Function:

```
SELECT SUM(SALARY) FROM EMPLOYEE
```

SUM(SALARY)

291000

```
SELECT SUM(SALARY) FROM EMPLOYEE WHERE CITY='PUNE'
```

SUM(SALARY)

51000

COUNT ( ) Function:

```
SELECT COUNT(ID) FROM EMPLOYEE
```

COUNT(ID)

8

COUNT (*) Function:

```
SELECT COUNT(*) FROM EMPLOYEE
```

COUNT(*)

8

```
SELECT JOB,COUNT(*) FROM EMPLOYEE GROUP BY JOB
```

JOB	COUNT(*)
CLECK	1
ANALYST	2
SALESMAN	2
CLERK	1
MANAGER	2

5 rows selected.

GROUP BY (Grouping Result)

```
SELECT JOB FROM EMPLOYEE GROUP BY JOB
```

JOB
CLECK
ANALYST
SALESMAN
CLERK
MANAGER

5 rows selected.

SQL CONSTRAINTS:

NOT NULL Constraint:

```
CREATE TABLE MYCONTACTS(
                    ID INT NOT NULL,
                    FNAME VARCHAR(225) NOT NULL,
                    LNAME VARCHAR(225) NOT NULL,
                    AGE INT,
                    CITY VARCHAR(225)
                    )
```

Table created.

```
INSERT INTO MYCONTACTS(ID,FNAME,LNAME) VALUES( 110,'ROMY','MITTAL')
```

1 row(s) inserted.

```
INSERT INTO MYCONTACTS(ID,FNAME,LNAME) VALUES( 110,'JOHN','WICCOT')
```

1 row(s) inserted.

```
INSERT INTO MYCONTACTS(ID,FNAME,LNAME,AGE) VALUES(
120,'MANIK','SHAH',25)
```

1 row(s) inserted.

```
SELECT * FROM MYCONTACTS
```

ID	FNAME	LNAME	AGE	CITY
110	ROMY	MITTAL	-	-
120	MANIK	SHAH	25	-
110	JOHN	WICCOT	-	-

3 rows selected.

```
UPDATE MYCONTACTS
SET AGE=35
WHERE AGE IS NULL
```

2 row(s) updated.

```
SELECT * FROM MYCONTACTS WHERE AGE IS NOT NULL
```

ID	FNAME	LNAME	AGE	CITY
110	ROMY	MITTAL	35	-

120 MANIK SHAH 25 -

110 JOHN WICCOT 35 -

3 rows selected.

Using ALTER

```
ALTER TABLE MYCONTACTS MODIFY AGE INT NOT NULL
```

Table altered.

```
INSERT INTO MYCONTACTS(ID,FNAME,LNAME) VALUES( 146,'DEEP','MICCOT')
```

ORA-01400: cannot insert NULL into ("SQL_WDIPPFJSGRWBCNJQRPSLPIOLS"."MYCONTACTS"."AGE") ORA-06512: at "SYS.DBMS_SQL", line 1721

EXPLANATION: Age cannot have null value because we are using NOT NULL constraint

UNIQUE Constraints:

```
CREATE TABLE MYCONTACTS(
                  ID INT UNIQUE,
                  FNAME VARCHAR(225) NOT NULL,
                  LNAME VARCHAR(225) NOT NULL,
                  AGE INT DEFAULT 20,
                  CITY VARCHAR(225)
                  )
```

Table created.

```
INSERT INTO MYCONTACTS(ID,FNAME,LNAME,AGE) VALUES(
110,'ROMY','MITTAL',26)
```

1 row(s) inserted.

```
INSERT INTO MYCONTACTS(ID,FNAME,LNAME,AGE) VALUES(
120,'MANIK','SHAH',25)
```

1 row(s) inserted.

```
INSERT INTO MYCONTACTS(ID,FNAME,LNAME,AGE,CITY) VALUES(
170,'JOHN','WICCOT',23,'MUMBAI')
```

1 row(s) inserted.

```
INSERT INTO MYCONTACTS(ID,FNAME,LNAME,CITY) VALUES(
150,'TOMY','NITTAL','PUNJAB')
```

1 row(s) inserted.

```
SELECT * FROM MYCONTACTS
```

ID	FNAME	LNAME	AGE	CITY
120	MANIK	SHAH	25	-
170	JOHN	WICCOT	23	MUMBAI
150	TOMY	NITTAL	20	PUNJAB
110	ROMY	MITTAL	26	-

4 rows selected.

Using ALTER

```
ALTER TABLE MYCONTACTS ADD UNIQUE(AGE)
```

Table altered.

```
INSERT INTO MYCONTACTS(ID,FNAME,LNAME,AGE) VALUES(
180,'MFGIK','SHAH',25)
```

ORA-00001: unique constraint (SQL_WDIPPFJSGRWBCNJQRPSLPIOLS.SYS_C0058224263) violated ORA-06512: at "SYS.DBMS_SQL", line 1721

EXPLANATION: Age column should have unique values because we are using unique constraint

PRIMARY KEY Constraints:

```
CREATE TABLE MYCONTACTS(
                    ID INT PRIMARY KEY,
                    FNAME VARCHAR(225) NOT NULL,
                    LNAME VARCHAR(225) NOT NULL,
                    AGE INT,
                    CITY VARCHAR(225)
                    )
```

Table created.

```
INSERT INTO MYCONTACTS(ID,FNAME,LNAME,AGE) VALUES(
110,'ROMY','MITTAL',26)
```

1 row(s) inserted.

```
INSERT INTO MYCONTACTS(ID,FNAME,LNAME,AGE,CITY) VALUES(
170,'JOHN','WICCOT',23,'MUMBAI')
1 row(s) inserted.
INSERT INTO MYCONTACTS(ID,FNAME,LNAME,CITY) VALUES(
150,'TOMY','NITTAL','PUNJAB')
1 row(s) inserted.
```

```
SELECT * FROM MYCONTACTS
```

ID	FNAME	LNAME	AGE	CITY
110	ROMY	MITTAL	26	-
170	JOHN	WICCOT	23	MUMBAI
150	TOMY	NITTAL	-	PUNJAB

3 rows selected.

Using ALTER

```
ALTER TABLE MYCONTACTS DROP PRIMARY KEY
Table altered.
```

```
ALTER TABLE MYCONTACTS ADD PRIMARY KEY(ID,LNAME,AGE)
Table altered.
```

```
INSERT INTO MYCONTACTS(ID,FNAME,LNAME,AGE) VALUES(
110,'MANIK','MITTAL',26)
```

ORA-00001: unique constraint (SQL_RDNRFQQSURPAMAABCZVKCXWWE.SYS_C0058291428) violated ORA-06512: at "SYS.DBMS_SQL", line 1721

EXPLANATION: The primary key constraint uniquely identifies each record in a table.

FOREIGN KEY Constraint:

```
CREATE TABLE PRODUCT(
                PID NUMBER PRIMARY KEY,
                PNAME VARCHAR(30),
                SNAME VARCHAR(30),
                UPRICE NUMBER
                )
```

Table created.

```
INSERT INTO PRODUCT VALUES(101,'TV','ONIDA',150)
```
1 row(s) inserted.
```
INSERT INTO PRODUCT VALUES(102,'FRIDGE','SAMSUNG',450)
```
1 row(s) inserted.
```
INSERT INTO PRODUCT VALUES(103,'IPOD','APPLE',75)
```
1 row(s) inserted.

```
SELECT * FROM PRODUCT
```

PID	PNAME	SNAME	UPRICE
102	FRIDGE	SAMSUNG	450
101	TV	ONIDA	150
103	IPOD	APPLE	75

3 rows selected.

```
CREATE TABLE ORDERS(
               OID NUMBER,
               PID NUMBER,
               TUNITS NUMBER,
               CUST VARCHAR(30)
               )
```
Table created.
```
INSERT INTO ORDERS VALUES(5101,102,25,'INFOSYS')
```
1 row(s) inserted.
```
INSERT INTO ORDERS VALUES(5102,103,5,'SATYAM')
```
1 row(s) inserted.
```
INSERT INTO ORDERS VALUES(5103,101,10,'TCS')
```
1 row(s) inserted.

```
SELECT * FROM ORDERS
```

OID	PID	TUNITS	CUST
5103	101	10	TCS
5102	103	5	SATYAM
5101	102	25	INFOSYS

3 rows selected.

Using ALTER

```
ALTER TABLE ORDERS ADD FOREIGN KEY(PID) REFERENCES PRODUCT(PID)
```

Table altered.

```
INSERT INTO ORDERS VALUES(5100,104,30,'WIPRO')
```

ORA-02291: integrity constraint (SQL_VTTTNLDPUYHFJYGRKUNOWITKT.SYS_C0058296906) violated - parent key not found ORA-06512: at "SYS.DBMS_SQL", line 1721

EXPLANATION: The `FOREIGN KEY` constraint is used to prevent actions that would destroy links between tables.

DEFAULT Constraints:

```
CREATE TABLE MYCONTACTS(
                ID INT NOT NULL,
                FNAME VARCHAR(225) NOT NULL,
                LNAME VARCHAR(225) NOT NULL,
                AGE INT DEFAULT 20,
                CITY VARCHAR(225)
                )
```

Table created.

```
INSERT INTO MYCONTACTS(ID,FNAME,LNAME) VALUES( 110,'ROMY','MITTAL')
```

1 row(s) inserted.

```
INSERT INTO MYCONTACTS(ID,FNAME,LNAME) VALUES( 110,'JOHN','WICCOT')
```

1 row(s) inserted.

```
INSERT INTO MYCONTACTS(ID,FNAME,LNAME,AGE) VALUES(
120,'MANIK','SHAH',25)
```

1 row(s) inserted.

```
SELECT * FROM MYCONTACTS
```

ID	FNAME	LNAME	AGE	CITY
110	ROMY	MITTAL	20	-
110	JOHN	WICCOT	20	-
120	MANIK	SHAH	25	-

3 rows selected.

Using ALTER

```
ALTER TABLE MYCONTACTS MODIFY CITY DEFAULT 'PARIS'
```

Table altered.

```
INSERT INTO MYCONTACTS(ID,FNAME,LNAME) VALUES( 150,'TOMY','NITTAL')
```

1 row(s) inserted.

```
SELECT * FROM MYCONTACTS
```

ID	FNAME	LNAME	AGE	CITY
110	ROMY	MITTAL	20	-
150	TOMY	NITTAL	20	PARIS
110	JOHN	WICCOT	20	-
120	MANIK	SHAH	25	-

4 rows selected.

EXPLANATION: The `DEFAULT` constraint is used to set a default value for a column.

CHECK Constraint:

```
CREATE TABLE MYCONTACTS(
                    ID INT,
                    FNAME VARCHAR(225) NOT NULL,
                    LNAME VARCHAR(225) NOT NULL,
                    AGE INT CHECK(Age>=18),
                    CITY VARCHAR(225)
                    )
```

Table created.

Using ALTER

```
ALTER TABLE MYCONTACTS ADD CHECK(ID<5)
```

Table altered.

```
INSERT INTO MYCONTACTS(ID,FNAME,LNAME,AGE) VALUES(
1,'ROMY','MITTAL',26)
```

1 row(s) inserted.

```
INSERT INTO MYCONTACTS(ID,FNAME,LNAME,AGE) VALUES(
3,'MANIK','SHAH',24)
```

1 row(s) inserted.

```
INSERT INTO MYCONTACTS(ID,FNAME,LNAME,AGE,CITY) VALUES(
5,'JOHN','WICCOT',18,'MUMBAI')
```

1 row(s) inserted.

```
SELECT * FROM MYCONTACTS
```

ID	FNAME	LNAME	AGE	CITY
3	MANIK	SHAH	24	-
1	ROMY	MITTAL	26	-
5	JOHN	WICCOT	18	MUMBAI

3 rows selected.

```
INSERT INTO MYCONTACTS(ID,FNAME,LNAME,AGE,CITY) VALUES(
8,'JOHN','WICCOT',18,'MUMBAI')
```

ORA-02290: check constraint (SQL_VTTTNLDPUYHFJYGRKUNOWITKT.SYS_C0058298380) violated ORA-06512: at "SYS.DBMS_SQL", line 1721

EXPLANATION: The `CHECK` constraint is used to limit the value range that can be placed in a column.

TABLE Constraints:

- When a constraint is applied to a group of columns of the table, it is called a table constraint.
- Column constraint is defined along with the end of the column.
- Table constraints are defined at the end of the table.

```
CREATE TABLE EMPLOYEE(
                    EMID NUMBER,
                    ENAME VARCHAR2(30) ,
                    AGE NUMBER,
                    SALARY NUMBER,
                    DEPT VARCHAR2(30) NOT NULL,
                    PRIMARY KEY(ENAME,DEPT)
                    )
```

Table created.

```
DESCRIBE EMPLOYEE
```

TABLE EMPLOYEE

Column	Null?	Type
EMID	-	NUMBER
ENAME	NOT NULL	VARCHAR2(30)
AGE	-	NUMBER
SALARY	-	NUMBER
DEPT	NOT NULL	VARCHAR2(30)

5 rows selected.

JOINS

INNER EQUI JOIN:

```
CREATE TABLE PRODUCT(
                PID NUMBER PRIMARY KEY,
                PNAME VARCHAR(30),
                SNAME VARCHAR(30),
                UPRICE NUMBER
                )
```

Table created.

```
INSERT INTO PRODUCT VALUES(101,'TV','ONIDA',150)
```

1 row(s) inserted.

```
INSERT INTO PRODUCT VALUES(102,'FRIDGE','SAMSUNG',450)
```

1 row(s) inserted.

```
INSERT INTO PRODUCT VALUES(103,'IPOD','APPLE',75)
```

1 row(s) inserted.

```
INSERT INTO PRODUCT VALUES(104,'MOBILE','NOKIA',100)
```

1 row(s) inserted.

```
SELECT * FROM PRODUCT
```

PID	PNAME	SNAME	UPRICE
101	TV	ONIDA	150
102	FRIDGE	SAMSUNG	450
103	IPOD	APPLE	75
104	MOBILE	NOKIA	100

4 rows selected.

```
CREATE TABLE ORDERS(
                OID NUMBER,
                PID NUMBER,
                TUNITS NUMBER,
                CUST VARCHAR(30),
                FOREIGN KEY(PID) REFERENCES PRODUCT(PID)
                )
```

Table created.

```
INSERT INTO ORDERS VALUES(5100,104,30,'INFOSYS')
```

1 row(s) inserted.

```
INSERT INTO ORDERS VALUES(5101,102,25,'SATYAM')
```

1 row(s) inserted.

```
INSERT INTO ORDERS VALUES(5102,103,5,'WIPRO')
```

1 row(s) inserted.

```
INSERT INTO ORDERS VALUES(5103,101,10,'TCS')
```

1 row(s) inserted.

```
SELECT * FROM ORDERS
```

OID	PID	TUNITS	CUST

5100	104	30	INFOSYS
5101	102	25	SATYAM
5102	103	5	WIPRO
5103	101	10	TCS

4 rows selected.

```
SELECT O.OID,P.PNAME,P.UPRICE,P.SNAME,O.TUNITS
FROM PRODUCT P,ORDERS O
WHERE O.PID=P.PID
```

OID	PNAME	UPRICE	SNAME	TUNITS
5100	MOBILE	100	NOKIA	30
5101	FRIDGE	450	SAMSUNG	25
5102	IPOD	75	APPLE	5
5103	TV	150	ONIDA	10

4 rows selected.