REVIEW-1

ITE-1004 – Database Management Systems

19BIT0300-Kota Nikhil
19BIT0349-Putta vamshi
19BIT0292-Bhaumik Tandan

Faculty: - BIMAL KUMAR RAY
Slot: - L19 + L20

INVENTORY MANAGEMENT SYSTEM

Mini World and Description

In the modern world everything is getting automated. All the tedious work that takes a lot of time manually, can be done in the blink of an eye using automation.

Commodities are being bought and sold on daily basis, automation in this field is the need of the hour.

This system keep tracks on raw materials used by a factory , employee details of a store , brand details and also the customer details.

This will offer different front ends for different kinds of users, and thus data security and data abstraction.

It offers control over redundancy and thus reduces data duplication.

Data Requirement

- Product is identified by its product id, along with it, we have name, warranty and price of the product.

- Store is identified by its store id, along with it we have its phone number , name and address.

- Employee is identified by his id along with we have his salary,email id and name.

- Customer is identified by his id along with we have his phone number, emal id and name.

- Categories is identified by its category id, along with it, we have it’s name and description.

- Brand is identified by brand id and along with we have it’s name and phone number.

- Warehouse is identified by its id and along with we have its phone number and address.

- Factory is identified by it’s id and along with we have its phone number, owner details, name of factory and address.

- Raw material is identified by it’s name along with we have it’s price suplier details.

Cardinality Relationship Type

- Product can be in more than one store and a store can have more than one products, but each store must have a product.

- We can have multiple products in a category but each product belongs to only one category, each product must have a category.

- Each product can be manufactured in one factories and but converse is not true and it is compulsory for each product to be manufactured in a factory.

- Each product must have a single brand but each brand can have many products, we also have the release date of the product by the brand.

- One employee must work for only one store whereas one store can have one or more employees.

- One customer must be in the customer records of only one store and one store can have more than one customers, we have the details of the date of the product on which it was added in the shop.

- The store can be 100% automated and also can have no customers.

- One factory must have one warehouse whereas one warehouse can be jointly owned.

- One factory can have many raw materials and converse, but each factory must have a raw material.

- Each employee can work for a type of product (category), one type of product can ave more than one employees working

Functional Requirement

REMOVAL OF DATA

- Removal of warehouse if it does not have a factory.
- Removal category if it does not have any products.
- Removal of brand if it does not launches any product.
- Removal of factory if it does not produces any product.
- Removal of raw material if it not used in any factory.

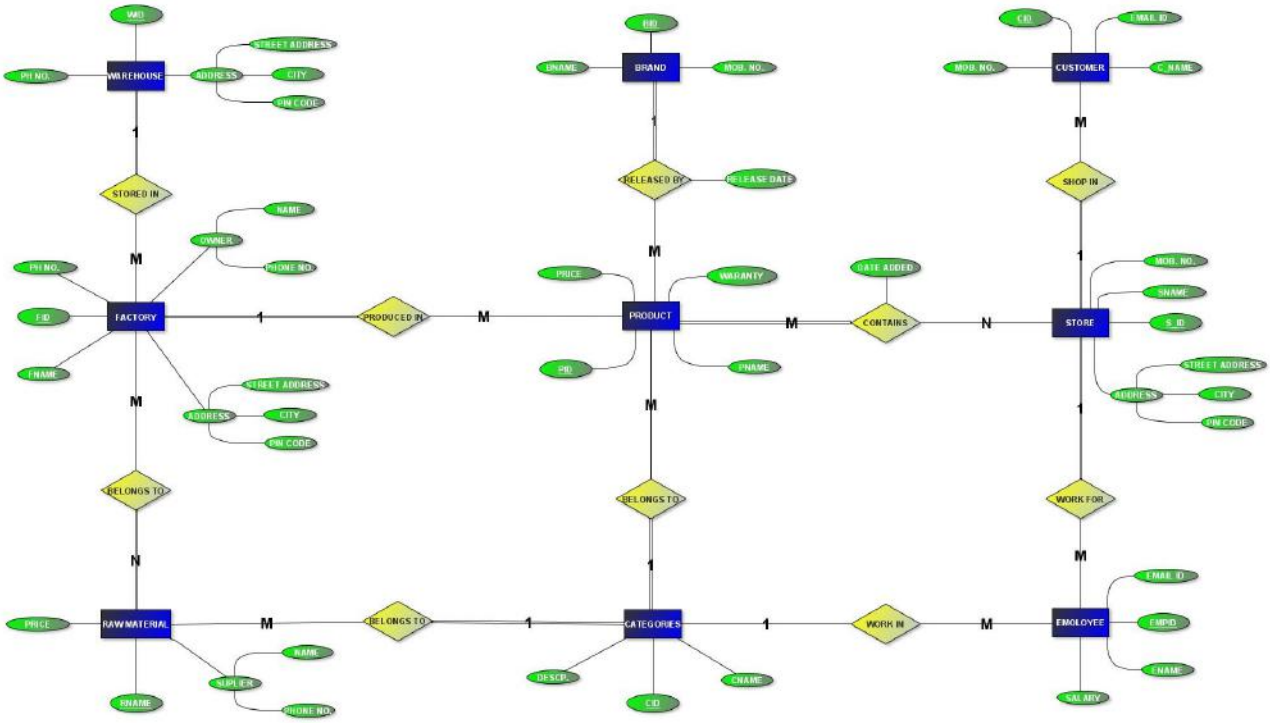
MODIFICATION OF DATA

- Update in the address of the warehouse if it is shifted.
- Update of phone number of the warehouse if it changes.
- Update of price of product and raw material according to market price.
- Update of phone number of brand and customer.
- Update the name of customer, brand, etc. in case it changes.

DATA RETRIEVAL

- Store can access the brand details of a product.
- Brand can view the details of the store in which their product is present.
- Brand can access the category details of a product.
- Factory can access the details about his warehouse and it’s raw materials.
- Customer can access the product details.
- Store can access the customer and employee details.
- Brand can access the factory details of their product.
- Brand can access the the customer details to directly contact them for the feedback of their product.

ER DIAGRAM


WID
WAREHOUSE
PH NO.
ADDRESS
STREET ADDRESS
CITY
PIN CODE
STORED IN
NAME
OWNER
PHONE NO.
FACTORY
FID
FNAME
PRODUCED IN
BELONGS TO
RAW MATERIAL
PRICE
RNAME
SUPPLIER
BELONGS TO
BID
BRAND
BNAME
MOB. NO.
RELEASED BY
RELEASE DATE
PRODUCT
PRICE
WARANTY
PID
PNAME
DATE ADDED
CONTAINS
BELONGS TO
CATEGORIES
DESCP.
CNAME
CID
WORK IN
CID
EMAIL ID
MOB. NO.
CUSTOMER
C_NAME
SHOP IN
MOB. NO.
SNAME
S_ID
STORE
STREET ADDRESS
ADDRESS
CITY
PIN CODE
WORK FOR
EMAIL ID
EMPID
ENAME
SALARY
EMOLOYEE

REVIEW-2

ITE-1004 – Database Management Systems

19BIT0300-Kota Nikhil
19BIT0349-Putta vamshi
19BIT0292-Bhaumik Tandan

Faculty: - BIMAL KUMAR RAY
Slot: - L19 + L20

INVENTORY MANAGEMENT SYSTEM

Relational Database Schema Diagram

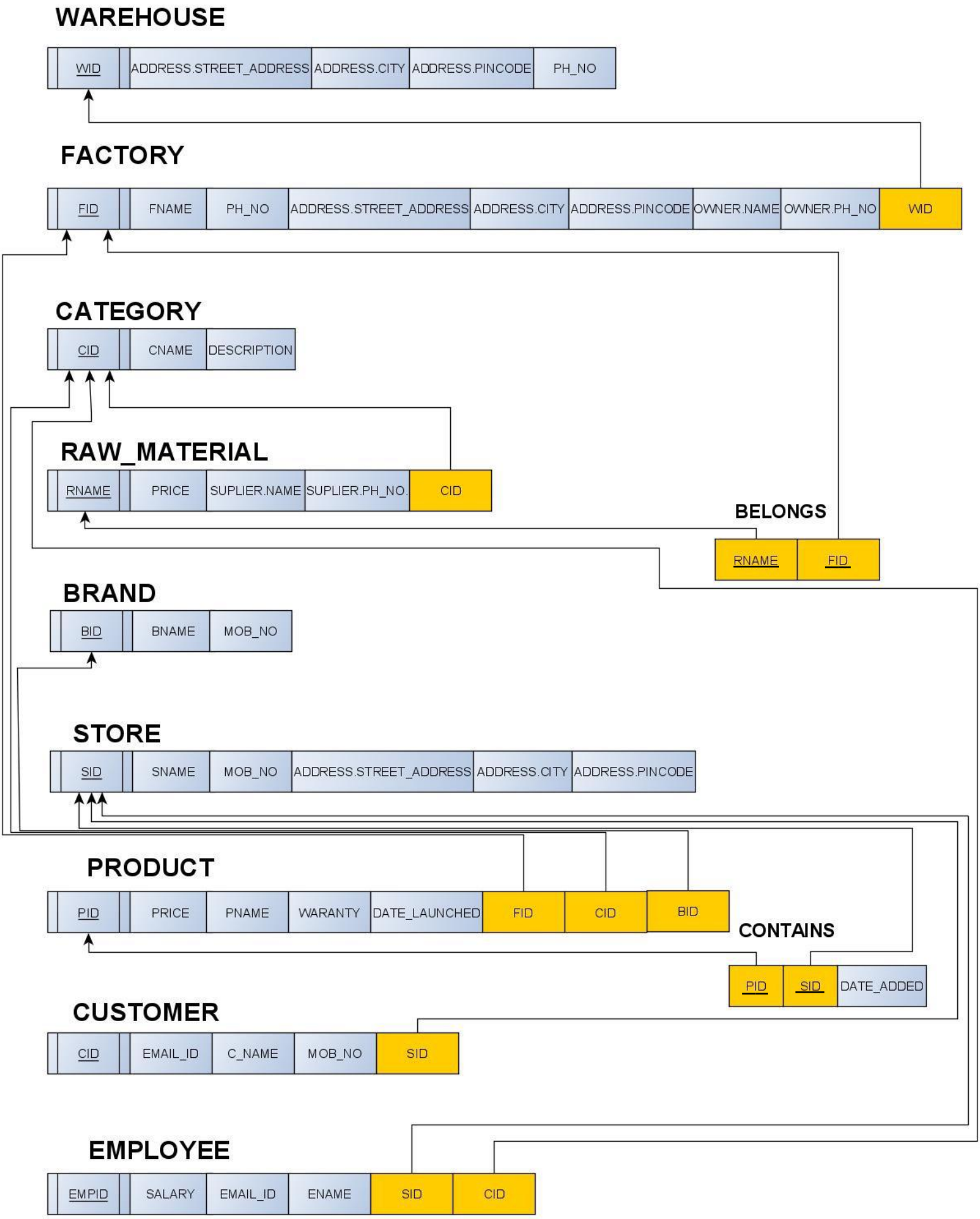


Table Creation

List of Tables

- WAREHOUSE
- FACTORY
- CATEGORY
- RAW_MATERIAL
- BELONGS
- BRAND
- STORE
- PRODUCT
- CONTAINS
- CUSTOMER
- EMPLOYEE

List of Abstract Data Type

- ADR(used for address)
- OWN (used for phone number and name of person)

Code for Table Creation

```
create type adr as object
(
  Street_address VARCHAR(50),
  city VARCHAR(15),
  pincode number(6)
);

CREATE type own as OBJECT
(
  name VARCHAR(15),
  ph_no number(10)
);

create table warehouse
(
  wid varchar(5) constraint p_wid PRIMARY key,
  address adr,
  PH_no varchar(10) constraint u_ph UNIQUE,
  CONSTRAINT u_sa UNIQUE(address.street_address),
  CONSTRAINT l_w CHECK((LENgth(PH_no)=10)AND(LENgth(wid)=5))
);

CREATE table factory
(
  fid varchar(5) CONSTRAINT p_fid PRIMARY KEY,
  fname VARCHAR(30) constraint n_fname NOT NULL,
  wid REFERENCES warehouse constraint f_k NOT NULL,
  ph_no number(10) CONSTRAINT u_pn unique,
  address adr,
  owner own,
  CONSTRAINT u_Fsa UNIQUE(address.street_address),
  constraint l_ph check((LENgth(PH_no)=10)AND(LENgth(fid
)=5)and(LENgth(owner.PH_no)=10))
);
```

```
CREATE table category
(
 cname varchar(20) constraint nn_cn NOT NULL,
 cid varchar(5) constraint p_cid PRIMARY KEY,
 des_ varchar(50),
 CONSTRAINT lcid CHECK (length(cid)=5)
);

CREATE table raw_material
(
 rname varchar(20) constraint u_n primary key,
 price number(6,2),
 suplier own,
 cid REFERENCES category CONSTRAINT NN_CI NOT NULL
);
create table belongs
(
 rname REFERENCES raw_material,
 fid REFERENCES factory
);

create table brand
(
 bid varchar(5) CONSTRAINT p_bid PRIMARY KEY,
 bname varchar(20) constraint nn_n NOT NULL,
 Mob_no varchar(10) constraint uB_pn UNIQUE,
 constraint l_pi CHECK((LENgth(Mob_no)=10)AND(LENgth(bid)=5))
);

create table product
(
 pid varchar(5) constraint p_id PRIMARY KEY,
 price number(6,2) constraint nnp_p NOT NULL,
 pname varchar(30) constraint nn_np NOT NULL,
 release_d date,
 waranty number(2),
 fid REFERENCES factory constraint f_f not null,
 cid REFERENCES category constraint f_c not null,
 bid REFERENCES brand constraint f_b not null,
 constraint loid  CHECK(LENgth(pid)=5)
);
```

```
create table store
(
  s_id varchar(5) constraint p_Sid PRIMARY KEY,
  sname VARCHAR(15) constraint nn_Sn not null,
  Ph_no varchar(10) constraint u_Spn UNIQUE,
  address adr,
  constraint l_Spn CHECK(LENgth(ph_no)=10),
  CONSTRAINT u_Ssa UNIQUE(address.street_address)
);

CREATE table CONTAINS
(
  pid REFERENCES product constraint t_p NOT null,
  s_id REFERENCES store constraint t_st NOT null,
  date_added date
);

CREATE table employee
(
  eid varchar(5) constraint pE_id PRIMARY KEY,
  salary number(6,2),
  email_id varchar(30),
  ename varchar(20) constraint nnE_n NOT NULL,
  s_id REFERENCES store constraint fE_id NOT NULL,
  cid REFERENCES category,
  CONSTRAINT lE_id CHECK (LENgth(eid)=5)
);


CREATE table customer
(
  cid varchar(5) constraint pC_id PRIMARY KEY,
  email_id varchar(30),
  Mob_n varchar(10) constraint uC_pn UNIQUE,
  c_name varchar(20) constraint nnC_n NOT NULL,
  s_id REFERENCES store constraint fC_t NOT NULL,
  constraint lnn CHECK ((LENgth(Mob_n)=10)AND(LENgth(cid
)=5)and((email_id is not null)or(mob_n is not null)))
);
```

Run Code in Oracle

Type ADR

```
SQL> create type adr as object
  2  (
  3      Street_address VARCHAR(50),
  4      city VARCHAR(15),
  5      pincode number(6)
  6  );
  7  /

Type created.

SQL> DESC ADR;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 STREET_ADDRESS                                     VARCHAR2(50)
 CITY                                               VARCHAR2(15)
 PINCODE                                            NUMBER(6)

SQL>
```

Type OWN

```
SQL> CREATE type own as OBJECT
  2  (
  3      name VARCHAR(15),
  4      ph_no number(10)
  5  );
  6  /

Type created.

SQL> DESC OWN;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 NAME                                               VARCHAR2(15)
 PH_NO                                              NUMBER(10)

SQL>
```

TABLE WAREHOUSE

```
SQL> create table warehouse
  2  (
  3       wid varchar(5) constraint p_wid PRIMARY key,
  4       address adr,
  5       PH_no varchar(10) constraint u_ph UNIQUE,
  6       CONSTRAINT u_sa UNIQUE(address.street_address),
  7       CONSTRAINT l_w CHECK((LENgth(PH_no)=10)AND(LENgth(wid)=5))
  8  );

Table created.

SQL> desc warehouse
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 WID                                       NOT NULL VARCHAR2(5)
 ADDRESS                                            ADR
 PH_NO                                              VARCHAR2(10)

SQL>
```

TABLE FACTORY

```
SQL> CREATE table factory
  2  (
  3       fid varchar(5) CONSTRAINT p_fid PRIMARY KEY,
  4       fname VARCHAR(30) constraint n_fname NOT NULL,
  5       wid REFERENCES warehouse constraint f_k NOT NULL,
  6       ph_no number(10) CONSTRAINT u_pn unique,
  7       address adr,
  8       owner own,
  9       CONSTRAINT u_Fsa UNIQUE(address.street_address),
 10       constraint l_ph check((LENgth(PH_no)=10)AND(LENgth(fid)=5)and(LENgth(owner.PH_no)=10))
 11  );

Table created.

SQL> DESC FACTORY;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 FID                                       NOT NULL VARCHAR2(5)
 FNAME                                     NOT NULL VARCHAR2(30)
 WID                                       NOT NULL VARCHAR2(5)
 PH_NO                                              NUMBER(10)
 ADDRESS                                            ADR
 OWNER                                              OWN

SQL>
```

TABLE CATEGORY

```
SQL> CREATE table category
  2  (
  3      cname varchar(20) constraint nn_cn NOT NULL,
  4      cid varchar(5) constraint p_cid PRIMARY KEY,
  5      des_ varchar(50),
  6      CONSTRAINT lcid CHECK (length(cid)=5)
  7  );

Table created.

SQL> DESC CATEGORY;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 CNAME                                     NOT NULL VARCHAR2(20)
 CID                                       NOT NULL VARCHAR2(5)
 DES_                                               VARCHAR2(50)

SQL>
```

TABLE RAW_MATERIAL

```
SQL> CREATE table raw_material
  2  (
  3      rname varchar(20) constraint u_n primary key,
  4      price number(6,2),
  5      suplier own,
  6      cid REFERENCES category CONSTRAINT NN_CI NOT NULL
  7  );

Table created.

SQL> DESC RAW_MATERIAL;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 RNAME                                     NOT NULL VARCHAR2(20)
 PRICE                                              NUMBER(6,2)
 SUPLIER                                            OWN
 CID                                       NOT NULL VARCHAR2(5)

SQL>
```

TABLE BELONGS

```
SQL> create table belongs
  2  (
  3      rname REFERENCES raw_material,
  4      fid REFERENCES factory
  5  );

Table created.

SQL> DESC BELONGS;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 RNAME                                              VARCHAR2(20)
 FID                                                VARCHAR2(5)

SQL>
```

TABLE BRAND

```
SQL> create table brand
  2  (
  3      bid varchar(5) CONSTRAINT p_bid PRIMARY KEY,
  4      bname varchar(20) constraint nn_n NOT NULL,
  5      Mob_no varchar(10) constraint uB_pn UNIQUE,
  6      constraint l_pi CHECK((LENgth(Mob_no)=10)AND(LENgth(bid)=5))
  7  );

Table created.

SQL> DESC BRAND;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 BID                                       NOT NULL VARCHAR2(5)
 BNAME                                     NOT NULL VARCHAR2(20)
 MOB_NO                                             VARCHAR2(10)

SQL>
```

TABLE PRODUCT

```
SQL> create table product
  2  (
  3      pid varchar(5) constraint p_id PRIMARY KEY,
  4      price number(6,2) constraint nnp_p NOT NULL,
  5      pname varchar(30) constraint nn_np NOT NULL,
  6      release_d date,
  7      waranty number(2),
  8      fid REFERENCES factory constraint f_f not null,
  9      cid REFERENCES category constraint f_c not null,
 10      bid REFERENCES brand constraint f_b not null,
 11      constraint loid  CHECK(LENgth(pid)=5)
 12  );

Table created.

SQL> DESC PRODUCT;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 PID                                       NOT NULL VARCHAR2(5)
 PRICE                                     NOT NULL NUMBER(6,2)
 PNAME                                     NOT NULL VARCHAR2(30)
 RELEASE_D                                          DATE
 WARANTY                                            NUMBER(2)
 FID                                       NOT NULL VARCHAR2(5)
 CID                                       NOT NULL VARCHAR2(5)
 BID                                       NOT NULL VARCHAR2(5)

SQL>
```

TABLE STORE

```
SQL> create table store
  2  (
  3      s_id varchar(5) constraint p_Sid PRIMARY KEY,
  4      sname VARCHAR(15) constraint nn_Sn not null,
  5      Ph_no varchar(10) constraint u_Spn UNIQUE,
  6      address adr,
  7      constraint l_Spn CHECK(LENgth(ph_no)=10),
  8      CONSTRAINT u_Ssa UNIQUE(address.street_address)
  9  );

Table created.

SQL> DESC STORE;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 S_ID                                      NOT NULL VARCHAR2(5)
 SNAME                                     NOT NULL VARCHAR2(15)
 PH_NO                                              VARCHAR2(10)
 ADDRESS                                            ADR

SQL>
```

TABLE CONTAINS

```
SQL> CREATE table CONTAINS
  2  (
  3      pid REFERENCES product constraint t_p NOT null,
  4      s_id REFERENCES store constraint t_st NOT null,
  5      date_added date
  6  );

Table created.

SQL> DESC CONTAINS;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 PID                                       NOT NULL VARCHAR2(5)
 S_ID                                      NOT NULL VARCHAR2(5)
 DATE_ADDED                                         DATE

SQL>
```

TABLE EMPLOYEE

```
SQL> CREATE table employee
  2  (
  3      eid varchar(5) constraint pE_id PRIMARY KEY,
  4      salary number(6,2),
  5      email_id varchar(30),
  6      ename varchar(20) constraint nnE_n NOT NULL,
  7      s_id REFERENCES store constraint fE_id NOT NULL,
  8      cid REFERENCES category,
  9      CONSTRAINT lE_id CHECK (LENgth(eid)=5)
 10  );

Table created.

SQL> DESC EMPLOYEE;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 EID                                       NOT NULL VARCHAR2(5)
 SALARY                                             NUMBER(6,2)
 EMAIL_ID                                           VARCHAR2(30)
 ENAME                                     NOT NULL VARCHAR2(20)
 S_ID                                      NOT NULL VARCHAR2(5)
 CID                                                VARCHAR2(5)

SQL>
```

TABLE CUSTOMER

```
SQL> CREATE table customer
  2  (
  3      cid varchar(5) constraint pC_id PRIMARY KEY,
  4      email_id varchar(30),
  5      Mob_n varchar(10) constraint uC_pn UNIQUE,
  6      c_name varchar(20) constraint nnC_n NOT NULL,
  7      s_id REFERENCES store constraint fC_t NOT NULL,
  8      constraint lnn CHECK ((LENgth(Mob_n)=10)AND(LENgth(cid)=5)and((email_id is not null)or(mob_n is not null)))
  9  );

Table created.

SQL> DESC CUSTOMER;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 CID                                       NOT NULL VARCHAR2(5)
 EMAIL_ID                                           VARCHAR2(30)
 MOB_N                                              VARCHAR2(10)
 C_NAME                                    NOT NULL VARCHAR2(20)
 S_ID                                      NOT NULL VARCHAR2(5)

SQL>
```

DATA INSERTION

WAREHOUSE TABLE

CODE:-

```
INSERT INTO WAREHOUSE VALUES
(
  '24345',
  ADR('101 A','LUCKNOW','226001'),
  9800425380
);

INSERT INTO WAREHOUSE VALUES
(
  '54345',
  ADR('201 MG ROAD','DELHI','324134'),
  8800525380
);

INSERT INTO WAREHOUSE VALUES
(
  '84645',
  ADR('904 AKBER ROAD','KOLKATA','454134'),
  9900543380
);
```

```
SQL> INSERT INTO WAREHOUSE VALUES
  2  (
  3       '24345',
  4       ADR('101 A','LUCKNOW','226001'),
  5       9800425380
  6  );

1 row created.

SQL> INSERT INTO WAREHOUSE VALUES
  2  (
  3       '54345',
  4       ADR('201 MG ROAD','DELHI','324134'),
  5       8800525380
  6  );

1 row created.

SQL> INSERT INTO WAREHOUSE VALUES
  2  (
  3       '84645',
  4       ADR('904 AKBER ROAD','KOLKATA','454134'),
  5       9900543380
  6  );

1 row created.

SQL> INSERT INTO WAREHOUSE VALUES
  2  (
  3       '54645',
  4       ADR('456 CANNOD PLACE','DELHI','436134'),
  5       7890543380
  6  );

1 row created.

SQL> INSERT INTO WAREHOUSE VALUES
  2  (
  3       '54455',
  4       ADR('715 M BLOCK','VELLORE','245432'),
  5       6590543380
  6  );

1 row created.
```

```sql
INSERT INTO WAREHOUSE VALUES
(
  '54645',
  ADR('456 CANNOD PLACE','DELHI','436134'),
  7890543380
);

INSERT INTO WAREHOUSE VALUES
(
  '54455',
  ADR('715 M BLOCK','VELLORE','245432'),
  6590543380
);
```

```
SQL> select * from warehouse;

WID
-----
ADDRESS(STREET_ADDRESS, CITY, PINCODE)
--------------------------------------------------------------------------------
PH_NO
----------
24345
ADR('101 A', 'LUCKNOW', 226001)
9800425380

54345
ADR('201 MG ROAD', 'DELHI', 324134)
8800525380

WID
-----
ADDRESS(STREET_ADDRESS, CITY, PINCODE)
--------------------------------------------------------------------------------
PH_NO
----------

84645
ADR('904 AKBER ROAD', 'KOLKATA', 454134)
9900543380

54645
ADR('456 CANNOD PLACE', 'DELHI', 436134)

WID
-----
ADDRESS(STREET_ADDRESS, CITY, PINCODE)
--------------------------------------------------------------------------------
PH_NO
----------
7890543380

54455
ADR('715 M BLOCK', 'VELLORE', 245432)
6590543380
```

FACTORY TABLE

CODE:-

```
INSERT INTO FACTORY VALUES
(
  '45645',
  'CITY FACTORY',
  '54345',
  9087678907,
  ADR('34 AKBAR ROAD','DELHI',456534),
  OWN('ANTONY THOMAS',6543456787)
);
INSERT INTO FACTORY VALUES
(
  '45685',
  'TOWN FACTORY',
  '84645',
  8087678907,
  ADR('14-FT ROAD','KANPUR',456534),
  OWN('RAHUL GUPTA',9443456787)
);

INSERT INTO FACTORY VALUES
(
  '55645',
  'CHARITY HOUSE',
  '24345',
  7087678907,
  ADR('40-FIT ROAD','MUMBAI',226534),
  OWN('DHAVAL MAVANI',9843456787)
);

INSERT INTO FACTORY VALUES
(
  '95645',
  'DAL FACTORY',
  '84645',
  9083478907,
  ADR('19 KALIDAS MARG','DURGAPUR',126534),
  OWN('YASH KHANDELWAL',7043456787)
);
```

```
SQL> INSERT INTO FACTORY VALUES
  2  (
  3       '45645',
  4       'CITY FACTORY',
  5       '54345',
  6       9087678907,
  7       ADR('34 AKBAR ROAD','DELHI',456534),
  8       OWN('ANTONY THOMAS',6543456787)
  9  );

1 row created.

SQL>
SQL> INSERT INTO FACTORY VALUES
  2  (
  3       '45685',
  4       'TOWN FACTORY',
  5       '84645',
  6       8087678907,
  7       ADR('14-FT ROAD','KANPUR',456534),
  8       OWN('RAHUL GUPTA',9443456787)
  9  );

1 row created.

SQL>
SQL> INSERT INTO FACTORY VALUES
  2  (
  3       '55645',
  4       'CHARITY HOUSE',
  5       '24345',
  6       7087678907,
  7       ADR('40-FIT ROAD','MUMBAI',226534),
  8       OWN('DHAVAL MAVANI',9843456787)
  9  );

1 row created.
```

```
SQL>
SQL> INSERT INTO FACTORY VALUES
  2  (
  3       '95645',
  4       'DAL FACTORY',
  5       '84645',
  6       9083478907,
  7       ADR('19 KALIDAS MARG','DURGAPUR',126534),
  8       OWN('YASH KHANDELWAL',7043456787)
  9  );

1 row created.

SQL>
SQL> INSERT INTO FACTORY VALUES
  2  (
  3       '85645',
  4       'CHOCLATE FACTORY',
  5       '24345',
  6       7080978907,
  7       ADR('20 RACE COURSE ROAD','KOLKATA',906534),
  8       OWN('RAHUL KUMAR',8943456787)
  9  );

1 row created.

SQL>
```

```sql
INSERT INTO FACTORY VALUES
(
  '85645',
  'CHOCLATE FACTORY',
  '24345',
  7080978907,
  ADR('20 RACE COURSE ROAD','KOLKATA',906534),
  OWN('RAHUL KUMAR',8943456787)
);
```

```
SQL> select * from factory;

FID   FNAME                          WID        PH_NO
----- ------------------------------ ----- ----------
ADDRESS(STREET_ADDRESS, CITY, PINCODE)
--------------------------------------------------------------------------------
OWNER(NAME, PH_NO)
--------------------------------------------------------------------------------
45645 CITY FACTORY                   54345 9087678907
ADR('34 AKBAR ROAD', 'DELHI', 456534)
OWN('ANTONY THOMAS', 6543456787)

45685 TOWN FACTORY                   84645 8087678907
ADR('14-FT ROAD', 'KANPUR', 456534)
OWN('RAHUL GUPTA', 9443456787)

FID   FNAME                          WID        PH_NO
----- ------------------------------ ----- ----------
ADDRESS(STREET_ADDRESS, CITY, PINCODE)
--------------------------------------------------------------------------------
OWNER(NAME, PH_NO)
--------------------------------------------------------------------------------

55645 CHARITY HOUSE                  24345 7087678907
ADR('40-FIT ROAD', 'MUMBAI', 226534)
OWN('DHAVAL MAVANI', 9843456787)

95645 DAL FACTORY                    84645 9083478907
ADR('19 KALIDAS MARG', 'DURGAPUR', 126534)

FID   FNAME                          WID        PH_NO
----- ------------------------------ ----- ----------
ADDRESS(STREET_ADDRESS, CITY, PINCODE)
--------------------------------------------------------------------------------
OWNER(NAME, PH_NO)
--------------------------------------------------------------------------------
OWN('YASH KHANDELWAL', 7043456787)

85645 CHOCLATE FACTORY               24345 7080978907
ADR('20 RACE COURSE ROAD', 'KOLKATA', 906534)
OWN('RAHUL KUMAR', 8943456787)
```

CATEGORY TABLE

CODE:-

```
INSERT INTO CATEGORY VALUES
(
  'ELECTRONICS',
  '56545',
  'Voltage less than 5 volts'
);

INSERT INTO CATEGORY VALUES
(
  'Cosmetics',
  '78545',
  'Starts just at rupees 999'
);

INSERT INTO CATEGORY VALUES
(
  'Grocery',
  '56945',
  'Get anything from A to Z'
);

INSERT INTO CATEGORY(cname,cid) VALUES
(
  'Ladies',
  '34945'
);

INSERT INTO CATEGORY(cname,cid) VALUES
(
  'Kids',
  '99945'
);
```

```
SQL> INSERT INTO CATEGORY VALUES
  2  (
  3      'ELECTRONICS',
  4      '56545',
  5      'Voltage less than 5 volts'
  6  );

1 row created.

SQL>
SQL> INSERT INTO CATEGORY VALUES
  2  (
  3      'Cosmetics',
  4      '78545',
  5      'Starts just at rupees 999'
  6  );

1 row created.

SQL>
SQL> INSERT INTO CATEGORY VALUES
  2  (
  3      'Grocery',
  4      '56945',
  5      'Get anything from A to Z'
  6  );

1 row created.

SQL>
SQL> INSERT INTO CATEGORY(cname,cid) VALUES
  2  (
  3      'Ladies',
  4      '34945'
  5  );

1 row created.

SQL> INSERT INTO CATEGORY(cname,cid) VALUES
  2  (
  3      'Kids',
  4      '99945'
  5  );

1 row created.
```

```
SQL> select * from category;

CNAME                CID   DES_
-------------------- ----- --------------------------
ELECTRONICS          56545 Voltage less than 5 volts
Cosmetics            78545 Starts just at rupees 999
Grocery              56945 Get anything from A to Z
Ladies               34945
Kids                 99945
```

RAW_MATERIAL TABLE

CODE:-

```
INSERT INTO RAW_MATERIAL VALUES
(
  'WOOD',
  2898.34,
  OWN('NALIN GUPTA',8987389098
),
  '56945'
);

INSERT INTO RAW_MATERIAL(RNAM
E,PRICE,CID) VALUES
(
  'MOTHERBOARD',
  9898.34,
  '56545'
);

INSERT INTO RAW_MATERIAL VALU
ES
(
  'SUGARCANE',
  898.34,
  OWN('RAHUL YADAV',6545434567),
  '56945'
);
```

```
SQL> INSERT INTO RAW_MATERIAL VALUES
  2  (
  3      'WOOD',
  4      2898.34,
  5      OWN('NALIN GUPTA',8987389098),
  6      '56945'
  7  );

1 row created.

SQL>
SQL> INSERT INTO RAW_MATERIAL(RNAME,PRICE,CID) VALUES
  2  (
  3      'MOTHERBOARD',
  4      9898.34,
  5      '56545'
  6  );

1 row created.

SQL>
SQL> INSERT INTO RAW_MATERIAL VALUES
  2  (
  3      'SUGARCANE',
  4      898.34,
  5      OWN('RAHUL YADAV',6545434567),
  6      '56945'
  7  );

1 row created.

SQL>
SQL> INSERT INTO RAW_MATERIAL(RNAME,PRICE,CID) VALUES
  2  (
  3      'PULSES',
  4      98.34,
  5      '56945'
  6  );

1 row created.

SQL>
SQL> INSERT INTO RAW_MATERIAL(RNAME,PRICE,CID) VALUES
  2  (
  3      'COCA',
  4      198.34,
  5      '56945'
  6  );

1 row created.
```

```
INSERT INTO RAW_MATERIAL(RNAME,PRICE,CID) VA
LUES
(
  'PULSES',
  98.34,
  '56945'
);
```

```
INSERT INTO RAW_MATERIAL(RNAME,PRICE,CID) VA
LUES
(
  'COCA',
  198.34,
  '56945'
);
```

```
SQL> select * from raw_material;

RNAME                     PRICE
------------------- ----------
SUPLIER(NAME, PH_NO)
--------------------------------------------------------------------------------
CID
-----
WOOD                    2898.34
OWN('NALIN GUPTA', 8987389098)
56945

MOTHERBOARD             9898.34

56545

RNAME                     PRICE
------------------- ----------
SUPLIER(NAME, PH_NO)
--------------------------------------------------------------------------------
CID
-----

SUGARCANE                898.34
OWN('RAHUL YADAV', 6545434567)
56945

PULSES                    98.34


RNAME                     PRICE
------------------- ----------
SUPLIER(NAME, PH_NO)
--------------------------------------------------------------------------------
CID
-----
56945

COCA                     198.34

56945
```

BELONGS TABLE

CODE:-

```
INSERT INTO BELONGS VALUES
(
  'COCA',
  '45685'
);

INSERT INTO BELONGS VALUES
(
  'MOTHERBOARD',
  '45645'
);

INSERT INTO BELONGS VALUES
(
  'WOOD',
  '45685'
);

INSERT INTO BELONGS VALUES
(
  'PULSES',
  '55645'
);

INSERT INTO BELONGS VALUES
(
  'PULSES',
  '95645'
);

INSERT INTO BELONGS VALUES
(
  'COCA',
  '85645'
);
```

```
SQL> INSERT INTO BELONGS VALUES
  2  (
  3       'COCA',
  4       '45685'
  5  );

1 row created.

SQL>
SQL> INSERT INTO BELONGS VALUES
  2  (
  3       'MOTHERBOARD',
  4       '45645'
  5  );

1 row created.

SQL>
SQL> INSERT INTO BELONGS VALUES
  2  (
  3       'WOOD',
  4       '45685'
  5  );

1 row created.

SQL>
SQL> INSERT INTO BELONGS VALUES
  2  (
  3       'PULSES',
  4       '55645'
  5  );

1 row created.

SQL>
SQL> INSERT INTO BELONGS VALUES
  2  (
  3       'PULSES',
  4       '95645'
  5  );

1 row created.

SQL>
SQL> INSERT INTO BELONGS VALUES
  2  (
  3       'COCA',
  4       '85645'
  5  );

1 row created.

SQL>
```

```
SQL> select * from belongs;

RNAME                FID
-------------------- -----
COCA                 45685
MOTHERBOARD          45645
WOOD                 45685
PULSES               55645
PULSES               95645
COCA                 85645

6 rows selected.
```

BRAND TABLE

CODE:-

```
INSERT INTO BRAND VALUES
(
  '34343',
  'GOLDEN HARVEST',
  '9876340756'
);

INSERT INTO BRAND VALUES
(
  '94343',
  'UTTAM',
  '9096340756'
);

INSERT INTO BRAND VALUES
(
  '74343',
  'TATA',
  '9099040756'
);

INSERT INTO BRAND VALUES
(
  '90343',
  'RELIANCE',
  '8909340756'
);

INSERT INTO BRAND VALUES
(
  '89043',
  'BIRLA',
  '6789040756'
);
```

```
SQL> INSERT INTO BRAND VALUES
  2  (
  3      '34343',
  4      'GOLDEN HARVEST',
  5      '9876340756'
  6  );

1 row created.

SQL>
SQL> INSERT INTO BRAND VALUES
  2  (
  3      '94343',
  4      'UTTAM',
  5      '9096340756'
  6  );

1 row created.

SQL>
SQL> INSERT INTO BRAND VALUES
  2  (
  3      '74343',
  4      'TATA',
  5      '9099040756'
  6  );

1 row created.

SQL>
SQL> INSERT INTO BRAND VALUES
  2  (
  3      '90343',
  4      'RELIANCE',
  5      '8909340756'
  6  );

1 row created.

SQL>
SQL> INSERT INTO BRAND VALUES
  2  (
  3      '89043',
  4      'BIRLA',
  5      '6789040756'
  6  );

1 row created.

SQL>
```

```
SQL> select * from brand;

BID   BNAME                MOB_NO
----- -------------------- ----------
34343 GOLDEN HARVEST       9876340756
94343 UTTAM                9096340756
74343 TATA                 9099040756
90343 RELIANCE             8909340756
89043 BIRLA                6789040756
```

PRODUCT TABLE

CODE:-

```
INSERT INTO PRODUCT VALUES
(
  '45432',
  1000.34,
  'SAMPAN DAL',
  '02-AUG-2007',
  2,
  '95645',
  '56945',
  '34343'
);
```

```
INSERT INTO PRODUCT VALUES
(
  '45332',
  9454.34,
  'SUPERFAST COMPUTER',
  '06-OCT-2015',
  10,
  '45685',
  '56545',
  '90343'
);
```

```
SQL> INSERT INTO PRODUCT VALUES
  2  (
  3       '45432',
  4       1000.34,
  5       'SAMPAN DAL',
  6       '02-AUG-2007',
  7       2,
  8       '95645',
  9       '56945',
 10       '34343'
 11  );

1 row created.

SQL>
SQL> INSERT INTO PRODUCT VALUES
  2  (
  3       '45332',
  4       9454.34,
  5       'SUPERFAST COMPUTER',
  6       '06-OCT-2015',
  7       10,
  8       '45685',
  9       '56545',
 10       '90343'
 11  );

1 row created.
```

```
INSERT INTO PRODUCT(PID,PRICE,PNAME,RELEASE_D,FID,CID,BID) VALUES
(
  '55332',
  20.5,
  'DAIRY MILK',
  '05-DEC-2017',
  '85645',
  '99945',
  '74343'
);
```

```
INSERT INTO PRODUCT VALUES
(
  '45334',
  500.4,
  'SPONGY SHOES',
  '26-JUN-2017',
  2,
  '55645',
  '99945',
  '74343'
);
```

```
SQL> INSERT INTO PRODUCT(PID,PRICE,PNAME,RELEASE_D,FID,CID,BID) VALUES
  2  (
  3      '55332',
  4      20.5,
  5      'DAIRY MILK',
  6      '05-DEC-2017',
  7      '85645',
  8      '99945',
  9      '74343'
 10  );

1 row created.

SQL>
SQL> INSERT INTO PRODUCT VALUES
  2  (
  3      '45334',
  4      500.4,
  5      'SPONGY SHOES',
  6      '26-JUN-2017',
  7      2,
  8      '55645',
  9      '99945',
 10      '74343'
 11  );

1 row created.

SQL>
SQL> INSERT INTO PRODUCT VALUES
  2  (
  3      '45382',
  4      2000.33,
  5      'WOMEN SLEEPER',
  6      '25-NOV-2019',
  7      1,
  8      '55645',
  9      '34945',
 10      '90343'
 11  );

1 row created.
```

```sql
INSERT INTO PRODUCT VALUES
(
  '45382',
  2000.33,
  'WOMEN SLEEPER',
  '25-NOV-2019',
  1,
  '55645',
  '34945',
  '90343'
);
```

```
SQL> select * from product;

PID        PRICE PNAME                          RELEASE_D    WARANTY FID   CID
----- ---------- ------------------------------ --------- ---------- ----- -----
BID
-----
45432    1000.34 SAMPAN DAL                     02-AUG-07          2 95645 56945
34343

45332    9454.34 SUPERFAST COMPUTER             06-OCT-15         10 45685 56545
90343

55332       20.5 DAIRY MILK                     05-DEC-17            85645 99945
74343


PID        PRICE PNAME                          RELEASE_D    WARANTY FID   CID
----- ---------- ------------------------------ --------- ---------- ----- -----
BID
-----
45334      500.4 SPONGY SHOES                   26-JUN-17          2 55645 99945
74343

45382    2000.33 WOMEN SLEEPER                  25-NOV-19          1 55645 34945
90343
```

STORE TABLE

CODE:-

```
INSERT INTO STORE VALUES
(
  '89098',
  'BIG BAZAAR',
  '8789098767',
  ADR('23 WAY LANE SAHARAGANJ','LUCKNOW',226001)
);
```

```
INSERT INTO STORE VALUES
(
  '89008',
  'V-MART',
  '9889098767',
  ADR('33-KALIDAAS MARG','MUMBAI',256001)
);
```

```
INSERT INTO STORE VALUES
(
  '89308',
  'FOOD MALL',
  '6889098767',
  ADR('9-MG ROAD','KOLKATA',454001)
);
```

```
SQL> INSERT INTO STORE VALUES
  2  (
  3      '89098',
  4      'BIG BAZAAR',
  5      '8789098767',
  6      ADR('23 WAY LANE SAHARAGANJ','LUCKNOW',226001)
  7  );

1 row created.

SQL>
SQL> INSERT INTO STORE VALUES
  2  (
  3      '89008',
  4      'V-MART',
  5      '9889098767',
  6      ADR('33-KALIDAAS MARG','MUMBAI',256001)
  7  );

1 row created.

SQL>
SQL> INSERT INTO STORE VALUES
  2  (
  3      '89308',
  4      'FOOD MALL',
  5      '6889098767',
  6      ADR('9-MG ROAD','KOLKATA',454001)
  7  );

1 row created.
```

```
INSERT INTO STORE VALUES
(
  '99308',
  'RELAINCE TRENDS',
  '6889098267',
  ADR('2-
GT ROAD','BANGLORE',3440
01)
);
```

```
SQL>
SQL> INSERT INTO STORE VALUES
  2  (
  3       '99308',
  4       'RELAINCE TRENDS',
  5       '6889098267',
  6       ADR('2-GT ROAD','BANGLORE',344001)
  7  );

1 row created.

SQL>
SQL> INSERT INTO STORE VALUES
  2  (
  3       '98308',
  4       'WESTSIDE',
  5       '8089098267',
  6       ADR('K-20 KIDWAI NAGAR','KANPUR',444001)
  7  );

1 row created.

SQL> _
```

```
INSERT INTO STORE VALUES
(
  '98308',
  'WESTSIDE',
  '8089098267',
  ADR('K-20 KIDWAI NAGAR','KANPUR',444001)
);
```

```
SQL> select * from store;

S_ID  SNAME           PH_NO
----- --------------- ----------
ADDRESS(STREET_ADDRESS, CITY, PINCODE)
--------------------------------------------------
89098 BIG BAZAAR      8789098767
ADR('23 WAY LANE SAHARAGANJ', 'LUCKNOW', 226001)

89008 V-MART          9889098767
ADR('33-KALIDAAS MARG', 'MUMBAI', 256001)

89308 FOOD MALL       6889098767
ADR('9-MG ROAD', 'KOLKATA', 454001)


S_ID  SNAME           PH_NO
----- --------------- ----------
ADDRESS(STREET_ADDRESS, CITY, PINCODE)
--------------------------------------------------
99308 RELAINCE TRENDS 6889098267
ADR('2-GT ROAD', 'BANGLORE', 344001)

98308 WESTSIDE        8089098267
ADR('K-20 KIDWAI NAGAR', 'KANPUR', 444001)
```

CONTAINS TABLE

CODE:-

```
INSERT INTO CONTAINS VALUES
(
  '55332',
  '89098',
  '23-AUG-2020'
);

INSERT INTO CONTAINS VALUES
(
  '45432',
  '89098',
  '13-NOV-2019'
);

INSERT INTO CONTAINS VALUES
(
  '55332',
  '89008',
  '25-AUG-2020'
);

INSERT INTO CONTAINS VALUES
(
  '45432',
  '89308',
  '25-AUG-2019'
);
```

```
SQL> INSERT INTO CONTAINS VALUES
  2  (
  3       '55332',
  4       '89098',
  5       '23-AUG-2020'
  6  );

1 row created.

SQL>
SQL> INSERT INTO CONTAINS VALUES
  2  (
  3       '45432',
  4       '89098',
  5       '13-NOV-2019'
  6  );

1 row created.

SQL>
SQL> INSERT INTO CONTAINS VALUES
  2  (
  3       '55332',
  4       '89008',
  5       '25-AUG-2020'
  6  );

1 row created.

SQL>
SQL> INSERT INTO CONTAINS VALUES
  2  (
  3       '45432',
  4       '89308',
  5       '25-AUG-2019'
  6  );

1 row created.
```

```
INSERT INTO CONTAINS VALUES
(
  '55332',
  '89308',
  '27-AUG-2020'
);
```

```
INSERT INTO CONTAINS VALUES
(
  '45382',
  '99308',
  '07-SEP-2018'
);
```

```
INSERT INTO CONTAINS VALUES
(
  '45334',
  '98308',
  '17-JUL-2020'
);
```

```
INSERT INTO CONTAINS VALUES
(
  '55332',
  '98308',
  '17-SEP-2018'
);
```

```
SQL>
SQL> INSERT INTO CONTAINS VALUES
  2  (
  3       '55332',
  4       '89308',
  5       '27-AUG-2020'
  6  );

1 row created.

SQL>
SQL> INSERT INTO CONTAINS VALUES
  2  (
  3       '45382',
  4       '99308',
  5       '07-SEP-2018'
  6  );

1 row created.

SQL>
SQL> INSERT INTO CONTAINS VALUES
  2  (
  3       '45334',
  4       '98308',
  5       '17-JUL-2020'
  6  );

1 row created.

SQL>
SQL> INSERT INTO CONTAINS VALUES
  2  (
  3       '55332',
  4       '98308',
  5       '17-SEP-2018'
  6  );

1 row created.
```

```
SQL> select * from contains;

PID   S_ID  DATE_ADDE
----- ----- ---------
55332 89098 23-AUG-20
45432 89098 13-NOV-19
55332 89008 25-AUG-20
45432 89308 25-AUG-19
55332 89308 27-AUG-20
45382 99308 07-SEP-18
45334 98308 17-JUL-20
55332 98308 17-SEP-18

8 rows selected.
```

EMPLOYEE TABLE

CODE:-

```
INSERT INTO EMPLOYEE VALUES
(
  '35353',
  4345.43,
  'rahul.gupta@gmail.com',
  'RAHUL GUPTA',
  '89098',
  '56545'
);
```

```
INSERT INTO EMPLOYEE(EID,SALARY
,EMAIL_ID,ENAME,S_ID) VALUES
(
  '33353',
  5345.43,
  'rohin.gupta@gmail.com',
  'ROHIN GUPTA',
  '89098'
);
```

```
INSERT INTO EMPLOYEE(EID,SALARY,ENAME,S_ID,CID) V
ALUES
(
  '73033',
  9999.89,
  'NALIN GUPTA',
  '99308',
  '99945'
);
```

```
SQL> INSERT INTO EMPLOYEE VALUES
  2  (
  3      '35353',
  4      4345.43,
  5      'rahul.gupta@gmail.com',
  6      'RAHUL GUPTA',
  7      '89098',
  8      '56545'
  9  );

1 row created.

SQL>
SQL> INSERT INTO EMPLOYEE(EID,SALARY,EMAIL_ID,ENAME,S_ID) VALUES
  2  (
  3      '33353',
  4      5345.43,
  5      'rohin.gupta@gmail.com',
  6      'ROHIN GUPTA',
  7      '89098'
  8  );

1 row created.

SQL>
SQL> INSERT INTO EMPLOYEE(EID,SALARY,ENAME,S_ID,CID) VALUES
  2  (
  3      '73033',
  4      9999.89,
  5      'NALIN GUPTA',
  6      '99308',
  7      '99945'
  8  );

1 row created.
```

```sql
INSERT INTO EMPLOYEE(EID,ENAME,S_ID) VALUES
(
  '70033',
  'AKASH GUPTA',
  '99308'
);
```

```
SQL>
SQL> INSERT INTO EMPLOYEE(EID,ENAME,S_ID) VALUES
  2  (
  3      '70033',
  4      'AKASH GUPTA',
  5      '99308'
  6  );

1 row created.

SQL>
SQL> INSERT INTO EMPLOYEE VALUES
  2  (
  3      '70933',
  4      6899.89,
  5      'peku.sharma@gmail.com',
  6      'PEKU SHARMA',
  7      '89008',
  8      '78545'
  9  );

1 row created.

SQL> _
```

```sql
INSERT INTO EMPLOYEE VALUES
(
  '70933',
  6899.89,
  'peku.sharma@gmail.com',
  'PEKU SHARMA',
  '89008',
  '78545'
);
```

```
SQL> select * from employee;

EID       SALARY EMAIL_ID                    ENAME               S_ID  CID
----- ---------- ---------------------------- ------------------- ----- -----
35353    4345.43 rahul.gupta@gmail.com        RAHUL GUPTA         89098 56545
33353    5345.43 rohin.gupta@gmail.com        ROHIN GUPTA         89098
73033    9999.89                              NALIN GUPTA         99308 99945
70033                                         AKASH GUPTA         99308
70933    6899.89 peku.sharma@gmail.com        PEKU SHARMA         89008 78545
```

CUSTOMER TABLE

CODE:-

```
INSERT INTO CUSTOMER VALUES
(
  '34564',
  'rahul.sinha34@gmail.com',
  '6543445678',
  'RAHUL SINHA',
   '99308'
);
```

```
INSERT INTO CUSTOMER(CID,MOB_N,C_NAME,S_ID) VALUES
(
  '64564',
  '9043445678',
  'LOKESH SRIVASTAVA',
   '99308'
);
```

```
INSERT INTO CUSTOMER(CID,EMAIL_ID,C_NAME,S_ID) VALUES
(
  '84564',
  'manan.sharma@gmail.com',
  'MANAN SHARMA',
```

```
SQL> INSERT INTO CUSTOMER VALUES
  2  (
  3       '34564',
  4       'rahul.sinha34@gmail.com',
  5       '6543445678',
  6       'RAHUL SINHA',
  7        '99308'
  8  );

1 row created.

SQL>
SQL> INSERT INTO CUSTOMER(CID,MOB_N,C_NAME,S_ID) VALUES
  2  (
  3       '64564',
  4       '9043445678',
  5       'LOKESH SRIVASTAVA',
  6        '99308'
  7  );

1 row created.

SQL>
SQL> INSERT INTO CUSTOMER(CID,EMAIL_ID,C_NAME,S_ID) VALUES
  2  (
  3       '84564',
  4       'manan.sharma@gmail.com',
  5       'MANAN SHARMA',
  6        '89008'
  7  );

1 row created.
```

```
  '89008'
);
```

```
INSERT INTO CUSTOME
R VALUES
(
  '89564',
  'krishna.misra@gma
il.com',
  '9043445978',
  'KRISHNA MISRA',
  '89308'
);
```

```
SQL>
SQL> INSERT INTO CUSTOMER VALUES
  2  (
  3       '89564',
  4       'krishna.misra@gmail.com',
  5       '9043445978',
  6       'KRISHNA MISRA',
  7        '89308'
  8  );

1 row created.

SQL>
SQL> INSERT INTO CUSTOMER VALUES
  2  (
  3       '90964',
  4       'gopal.das@gmail.com',
  5       '6783445978',
  6       'GOPAL DAS',
  7        '89008'
  8  );

1 row created.

SQL>
```

```
INSERT INTO CUSTOMER VALUES
(
  '90964',
  'gopal.das@gmail.com',
  '6783445978',
  'GOPAL DAS',
  '89008'
);
```

```
SQL> select * from customer;

CID   EMAIL_ID                       MOB_N      C_NAME               S_ID
----- ------------------------------ ---------- -------------------- -----
34564 rahul.sinha34@gmail.com        6543445678 RAHUL SINHA          99308
64564                                9043445678 LOKESH SRIVASTAVA    99308
84564 manan.sharma@gmail.com                    MANAN SHARMA         89008
89564 krishna.misra@gmail.com        9043445978 KRISHNA MISRA        89308
90964 gopal.das@gmail.com            6783445978 GOPAL DAS            89008
```

ADDITION PRIMARY KEY USING ALTER

```
ALTER TABLE belongs

ADD CONSTRAINT at_cb

PRIMARY KEY (rname,fid);
```

```
SQL> ALTER TABLE belongs
  2  ADD CONSTRAINT at_cb
  3  PRIMARY KEY (rname,fid);

Table altered.
```

As each product can be added in a store multiple times on multiple dates so we need to include date in primary key also.

```
ALTER TABLE contains

ADD CONSTRAINT at_cc

PRIMARY KEY (pid,s_id,date_added);
```

```
SQL> ALTER TABLE contains
  2  ADD CONSTRAINT at_cc
  3  PRIMARY KEY (pid,s_id,date_added);

Table altered.
```

REVIEW-3

ITE-1004 – Database Management Systems

19BIT0300-Kota Nikhil
19BIT0349-Putta vamshi
19BIT0292-Bhaumik Tandan

Faculty: - BIMAL KUMAR RAY
Slot: - L19 + L20

INVENTORY MANAGEMENT SYSTEM

Queries

DELETION

a) Removal of warehouse if it does not have a factory. (query using set operator)

CODE

```
delete warehouse where wid in
(
  select wid from warehouse
      minus
  select wid from factory
);
```

```
SQL> delete warehouse where wid in
  2  (
  3       select wid from warehouse
  4              minus
  5       select wid from factory
  6  );

2 rows deleted.
```

b) Removal category if it does not have any products.

NOTE:- Since employee and raw material also have foreign key as category d so we have to first remove their category id if their category is of a category for which there is no product. So in spite of asling for data 3 times we save it in a emporary table.

CODE

(uncorrelated nested query)

```
create table NE
as
select cid from category
where cid not in
(
select cid from product
);
```

```
SQL> create table NE
  2       as
  3  select cid from category where
  4  cid not in
  5  (
  6       select cid from product
  7  );

Table created.

SQL> select * from ne;

CID
-----
78545
```

(correlated nested query using inner join)

```
update raw_material
rw set rw.cid=NULL
where EXISTS
(
 select * from ne inner join
          raw_material on
 raw_material.cid=ne.cid
 where
 rw.cid=raw_material.cid
);
```

```
SQL> update raw_material rw
  2  set rw.cid=NULL
  3  where EXISTS
  4  (
  5       select * from ne inner join raw_material
  6                              on
  7       raw_material.cid=ne.cid
  8       where
  9       rw.cid=raw_material.cid
 10  );

0 rows updated.
```

```
update employee em
set em.cid=NULL
where EXISTS
(
  select * from ne inner join employee
            on
  employee.cid=ne.cid
  where
  em.cid=employee.cid
);
```

```
SQL> update employee em
  2  set em.cid=NULL
  3  where EXISTS
  4  (
  5      select * from ne inner join employee
  6                          on
  7      employee.cid=ne.cid
  8      where
  9      em.cid=employee.cid
 10  );

1 row updated.
```

(query using natural join)

```
delete category where
cid IN
(
  select cid from ne natural join category
);
```

```
SQL> delete category where
  2  cid IN
  3  (
  4      select cid from ne natural join category
  5  );

1 row deleted.
```

NOTE:- Since now we do not need table ne drop that.

```
drop table ne;
```

```
SQL> drop table ne;

Table dropped.

SQL>
```

c) Removal of brand if it does not launches any product.

CODE

```
delete brand br WHERE
not EXISTS
(
  select bid from product
  where br.bid=bid
);
```

```
SQL> delete brand br WHERE
  2  not EXISTS
  3  (
  4       select bid from product
  5       where br.bid=bid
  6  );

2 rows deleted.
```

d) Removal of factory if it does not produces any product.

NOTE:- Since belongs table also have product id as foreign key so we also have to delete data from belongs table.

CODE

```
delete belongs
WHERE fid not IN
(
  select product.fid from
  factory inner join product
        on
  product.fid=factory.fid
);

delete factory
WHERE fid not IN
(
  select product.fid from
  factory inner join product
        on
  product.fid=factory.fid
);
```

```
SQL> delete belongs
  2  WHERE fid not IN
  3  (
  4       select product.fid from
  5       factory inner join product
  6                  on
  7       product.fid=factory.fid
  8  );

1 row deleted.

SQL> delete factory
  2  WHERE fid not IN
  3  (
  4       select product.fid from
  5       factory inner join product
  6                  on
  7       product.fid=factory.fid
  8  );

1 row deleted.
```

e) Removal of factory if it does not produces any product.

NOTE:- First we have to remove data from table belongs because it also have factory id as foreign key.

CODE

```
delete belongs where fid in
(
  select fid from factory
      minus
  select fid from product
);

delete factory where fid in
(
  select fid from factory
      minus
  select fid from product
);
```

```
SQL> delete belongs where fid in
  2  (
  3      select fid from factory
  4              minus
  5      select fid from product
  6  );

0 rows deleted.

SQL>
SQL> delete factory where fid in
  2  (
  3      select fid from factory
  4              minus
  5      select fid from product
  6  );

0 rows deleted.

SQL>
```

UPDATION

a)If the data of the salary of an employee is missing so set his salary to the average salary of the store in which they work.

DATA BEFORE UPDATION

```
SQL> select * from employee;

EID       SALARY EMAIL_ID                       ENAME               S_ID  CID
----- ---------- ------------------------------ ------------------- ----- -----
35353    4345.43 rahul.gupta@gmail.com          RAHUL GUPTA         89098 56545
33353    5345.43 rohin.gupta@gmail.com          ROHIN GUPTA         89098
73033    9999.89                                NALIN GUPTA         99308 99945
70033                                           AKASH GUPTA         99308
70933    6899.89 peku.sharma@gmail.com          PEKU SHARMA         89008
```

CODE

(using nvl, group by and having)

```
update employee em SET salary=
nvl(salary,
(
  select avg(salary) from
  employee group by s_id
  having s_id=em.s_id
)
);
```

```
SQL> SELECT * FROM EMPLOYEE WHERE SALARY=NULL;

no rows selected

SQL> update employee em SET salary=
  2  nvl(salary,
  3  (
  4      select avg(salary) from
  5      employee group by s_id
  6      having s_id=em.s_id
  7  )
  8  );

5 rows updated.

SQL> select * from employee;

EID       SALARY EMAIL_ID                       ENAME               S_ID  CID
----- ---------- ------------------------------ ------------------- ----- -----
35353    4345.43 rahul.gupta@gmail.com          RAHUL GUPTA         89098 56545
33353    5345.43 rohin.gupta@gmail.com          ROHIN GUPTA         89098
73033    9999.89                                NALIN GUPTA         99308 99945
70033    9999.89                                AKASH GUPTA         99308
70933    6899.89 peku.sharma@gmail.com          PEKU SHARMA         89008

SQL>
```

b) Increase the cost of all the products for a particular category for which a particular employee work.

DATA BEFORE UPDATION

```
SQL> select * from employee;

EID       SALARY EMAIL_ID                       ENAME               S_ID  CID
----- ---------- ------------------------------ ------------------- ----- -----
35353    4345.43 rahul.gupta@gmail.com          RAHUL GUPTA         89098 56545
33353    5345.43 rohin.gupta@gmail.com          ROHIN GUPTA         89098
73033    9999.89                                NALIN GUPTA         99308 99945
70033    9999.89                                AKASH GUPTA         99308
70933    6899.89 peku.sharma@gmail.com          PEKU SHARMA         89008

SQL> select * from category;

CNAME               CID   DES_
------------------- ----- --------------------------------------------------
ELECTRONICS         56545 Voltage less than 5 volts
Grocery             56945 Get anything from A to Z
Ladies              34945
Kids                99945

SQL> select * from product;

PID        PRICE PNAME                          RELEASE_D    WARANTY FID   CID
----- ---------- ------------------------------ --------- ---------- ----- -----
BID
-----
45432    1000.34 SAMPAN DAL                     02-AUG-07          2 95645 56945
34343

45332    9454.34 SUPERFAST COMPUTER             06-OCT-15         10 45685 56545
90343

55332       20.5 DAIRY MILK                     05-DEC-17            85645 99945
74343


PID        PRICE PNAME                          RELEASE_D    WARANTY FID   CID
----- ---------- ------------------------------ --------- ---------- ----- -----
BID
-----
45334      500.4 SPONGY SHOES                   26-JUN-17          2 55645 99945
74343

45382    2000.33 WOMEN SLEEPER                  25-NOV-19          1 55645 34945
90343
```

CODE

```
update product set price=
(price*(100+&per_incr))/100
where cid=(select cid from employee
where eid=&empid_for_increase);
```

OUTPUT

```
SQL> update product set price=
  2  (price*(100+&per_incr))/100
  3  where cid=(select cid from employee
  4  where eid=&empid_for_increase);
Enter value for per_incr: 10
old   2: (price*(100+&per_incr))/100
new   2: (price*(100+10))/100
Enter value for empid_for_increase: 73033
old   4: where eid=&empid_for_increase)
new   4: where eid=73033)

2 rows updated.

SQL> select * from product;

PID        PRICE PNAME                          RELEASE_D    WARANTY FID   CID
----- ---------- ------------------------------ --------- ---------- ----- -----
BID
-----
45432    1000.34 SAMPAN DAL                     02-AUG-07          2 95645 56945
34343

45332    9454.34 SUPERFAST COMPUTER             06-OCT-15         10 45685 56545
90343

55332      22.55 DAIRY MILK                     05-DEC-17            85645 99945
74343


PID        PRICE PNAME                          RELEASE_D    WARANTY FID   CID
----- ---------- ------------------------------ --------- ---------- ----- -----
BID
-----
45334     550.44 SPONGY SHOES                   26-JUN-17          2 55645 99945
74343

45382    2000.33 WOMEN SLEEPER                  25-NOV-19          1 55645 34945
90343
```

c) Change the address of a warehouse of the factory in which a particular product is made.

DATA BEFORE UPDATION

```
SQL> select fid,wid,pid from factory natural join product;

FID   WID   PID
----- ----- -----
45685 84645 45332
55645 24345 45334
55645 24345 45382
85645 24345 55332
95645 84645 45432

SQL> select wid,address from warehouse;

WID
-----
ADDRESS(STREET_ADDRESS, CITY, PINCODE)
--------------------------------------------------------
24345
ADR('101 A', 'LUCKNOW', 226001)

54345
ADR('201 MG ROAD', 'DELHI', 324134)

84645
ADR('904 AKBER ROAD', 'KOLKATA', 454134)
```

CODE

```
update warehouse set address=
adr('&street','&city',&pincode)
where wid=
(
  select wid from factory
    natural join product
  where pid=&give_pro_id
);
```

```
SQL> update warehouse set address=
  2  adr('&street','&city',&pincode)
  3  where wid=
  4  (
  5      select wid from factory
  6          natural join product
  7      where pid=&give_pro_id
  8  );
Enter value for street: 87 MG Road
Enter value for city: VELLORE
Enter value for pincode: 676787
old   2: adr('&street','&city',&pincode)
new   2: adr('87 MG Road','VELLORE',676787)
Enter value for give_pro_id: 45432
old   7:     where pid=&give_pro_id
new   7:     where pid=45432

1 row updated.

SQL> select address from warehouse
  2  where wid=84645;

ADDRESS(STREET_ADDRESS, CITY, PINCODE)
----------------------------------------
ADR('87 MG Road', 'VELLORE', 676787)
```

d) Increase the salary of all the employees which work for the category of a particular raw material

DATA BEFORE UPDATION

```
SQL> select cid,rname from category
  2  natural join raw_material;

CID   RNAME
----- --------------------
56945 WOOD
56545 MOTHERBOARD
56945 SUGARCANE
56945 PULSES
56945 COCA

SQL> select cid,eid,salary from employee;

CID   EID       SALARY
----- ----- ----------
56545 35353    4345.43
      33353    5345.43
99945 73033    9999.89
      70033    9999.89
      70933    6899.89
```

CODE

```
update employee set salary=
(salary*(100+&per_incr))/100
where cid=
(
  select cid from raw_material
  where rname='&raw_material'
);
```

```
SQL> update employee set salary=
  2  (salary*(100+&per_incr))/100
  3  where cid=
  4  (
  5      select cid from raw_material
  6      where rname='&raw_material'
  7  );
Enter value for per_incr: 30
old   2: (salary*(100+&per_incr))/100
new   2: (salary*(100+30))/100
Enter value for raw_material: MOTHERBOARD
old   6:     where rname='&raw_material'
new   6:     where rname='MOTHERBOARD'

1 row updated.
```

```
SQL> select eid,salary from employee;

EID       SALARY
----- ----------
35353    5649.06
33353    5345.43
73033    9999.89
70033    9999.89
70933    6899.89
```

DATA RETRIEVAL

a) Retrieve the brand details and product names for all the products that are made in a particular factory.

CODE

```
select pname,bid,bname,mob_no
from product natural join brand
where fid=&factory_id;
```

```
SQL> select pname,bid,bname,mob_no
  2  from product natural join brand
  3  where fid=&factory_id;
Enter value for factory_id: 55645
old   3: where fid=&factory_id
new   3: where fid=55645

PNAME                          BID   BNAME                MOB_NO
------------------------------ ----- -------------------- ----------
SPONGY SHOES                   74343 TATA                 9099040756
WOMEN SLEEPER                  90343 RELIANCE             8909340756
```

b) Retrieve the brand details of all the products other than than the brand detrail of a particular product.

DATA

```
SQL> select bid,pname,pid from product;

BID   PNAME                          PID
----- ------------------------------ -----
34343 SAMPAN DAL                     45432
90343 SUPERFAST COMPUTER             45332
74343 DAIRY MILK                     55332
74343 SPONGY SHOES                   45334
90343 WOMEN SLEEPER                  45382
```

CODE

(using nullif)

```
select * from brand where bid IN
(
  select bid from product
  where pid=nullif(pid,'&pid')
);
```

```
SQL> select * from brand where bid IN
  2  (
  3      select bid from product
  4      where pid=nullif(pid,'&pid')
  5  );
Enter value for pid: 45432
old   4:     where pid=nullif(pid,'&pid')
new   4:     where pid=nullif(pid,'45432')

BID   BNAME                MOB_NO
----- -------------------- ----------
74343 TATA                 9099040756
90343 RELIANCE             8909340756
```

c) Retrieve the customer details of all the customers that visit the stores in which the product of a particular brand is sold.

CODE

```
select * from customer
where s_id IN
(
  select s_id from CONTAINS
  where pid IN
  (
    select pid from PRODUCT
    where bid=&brand_id
  )
);
```

```
SQL> select * from customer
  2  where s_id IN
  3  (
  4      select s_id from CONTAINS
  5      where pid IN
  6      (
  7          select pid from PRODUCT
  8          where bid=&brand_id
  9      )
 10  );
Enter value for brand_id: 74343
old   8:         where bid=&brand_id
new   8:         where bid=74343

CID   EMAIL_ID                       MOB_N      C_NAME              S_ID
----- ------------------------------ ---------- ------------------- -----
84564 manan.sharma@gmail.com                    MANAN SHARMA        89008
89564 krishna.misra@gmail.com        9043445978 KRISHNA MISRA       89308
90964 gopal.das@gmail.com            6783445978 GOPAL DAS           89008
```

d) List the warehouse details of the factory in which the product of a brand is made.

CODE

```
select * from warehouse
where wid IN
(
  select distinct
  factory.wid
    from
  product join factory
       on
  product.fid=factory.fid
      where
  product.bid=&brand_id
);
```

```
SQL> select * from warehouse
  2  where wid IN
  3  (
  4       select distinct
  5       factory.wid
  6           from
  7       product join factory
  8                 on
  9       product.fid=factory.fid
 10              where
 11       product.bid=&brand_id
 12  );
Enter value for brand_id: 90343
old  11:      product.bid=&brand_id
new  11:      product.bid=90343

WID
-----
ADDRESS(STREET_ADDRESS, CITY, PINCODE)
----------------------------------------
PH_NO
---------
24345
ADR('101 A', 'LUCKNOW', 226001)
9800425380

84645
ADR('904 AKBER ROAD', 'KOLKATA', 454134)
9900543380
```

e) List the product name and the factory in which they are produced in alphabetical order of the product name.

CODE

```
select pname,fname from

product natural join

factory order by fname
;
```

```
SQL> select pname,fname from
  2  product natural join
  3  factory order by fname;

PNAME                          FNAME
------------------------------ ----------
WOMEN SLEEPER                  CHARITY HOUSE
SPONGY SHOES                   CHARITY HOUSE
DAIRY MILK                     CHOCLATE FACTORY
SAMPAN DAL                     DAL FACTORY
SUPERFAST COMPUTER             TOWN FACTORY
```

f) List the customer name and phone number of the store in which an employee work, if customer phone number is not present so print the phone number of the store.

DATA

```
SQL> select * from customer;

CID   EMAIL_ID                       MOB_N      C_NAME              S_ID
----- ------------------------------ ---------- ------------------- -----
34564 rahul.sinha34@gma              6543445678 RAHUL SINHA         99308
      il.com

64564                                9043445678 LOKESH SRIVASTAVA   99308
84564 manan.sharma@gmail.com                    MANAN SHARMA        89008
89564 krishna.misra@gmail.com        9043445978 KRISHNA MISRA       89308
90964 gopal.das@gmail.com            6783445978 GOPAL DAS           89008
```

```
SQL> select ph_no from store where s_id='89008';

PH_NO
----------
9889098767
```

CODE

(using nvl, correlated nested query and join)

```
select customer.c_name,nvl
(
  customer.mob_n,
  (
    select ph_no
    from store WHERE
    s_id=customer.s_id
  )
)  Phone_No
    from
customer join employee
    on
employee.s_id=customer.s_id
    where
employee.eid='&eid';
```

```
SQL> select customer.c_name,nvl
  2  (
  3      customer.mob_n,
  4      (
  5          select ph_no
  6          from store WHERE
  7          s_id=customer.s_id
  8      )
  9  )  Phone_No
 10          from
 11  customer join employee
 12            on
 13  employee.s_id=customer.s_id
 14           where
 15  employee.eid='&eid';
Enter value for eid: 70933
old  15: employee.eid='&eid'
new  15: employee.eid='70933'

C_NAME               PHONE_NO
-------------------- ----------
MANAN SHARMA         9889098767
GOPAL DAS            6783445978
```

G) Display store id and employee id of their corresponding employees of all the stores in the database, if an store does not have any employee print ‘automated’.

CODE

(using outer join and nvl)

```
select nvl(employee.eid,'AUTOMATED') EMP,store.s_id from store full outer join employee on employee.s_id=store.s_id;
```

```
SQL> select nvl(employee.eid,'AUTOMATED') EMP,store.s_id from store full outer join employee on employee.s_id=store.s_id;

EMP       S_ID
--------- -----
AUTOMATED 55555
70933     89008
33353     89098
35353     89098
AUTOMATED 89308
AUTOMATED 98308
70033     99308
73033     99308

8 rows selected.
```

FUNCTION CREATION

a) Create a function that accepts the store id of a store and return the total salary (rounded off to closest integer) of all the employees in that store, if the store does have any employee return null.

CODE

```
CREATE OR REPLACE FUNCTION
tosal (a in employee.s_id%type)
return int is
s number(8,2):=0;
cursor st is select salary
from employee where s_id=a;
sf st%rowtype;
BEGIN
open st;
fetch st into sf;
if st%rowcount=0 then
close st;
return null;
end if;
close st;
for sf in st loop
s:=s+sf.salary;
end loop;
return s;
end;
```

```
SQL> CREATE OR REPLACE FUNCTION
  2  tosal (a in employee.s_id%type)
  3  return int is
  4  s number(8,2):=0;
  5  cursor st is select salary
  6  from employee where s_id=a;
  7  sf st%rowtype;
  8  BEGIN
  9  open st;
 10  fetch st into sf;
 11  if st%rowcount=0 then
 12  close st;
 13  return null;
 14  end if;
 15  close st;
 16  for sf in st loop
 17  s:=s+sf.salary;
 18  end loop;
 19  return s;
 20  end;
 21  /

Function created.
```

OUTPUT

```
SQL> select s_id,sum(salary) from employee group by s_id;

S_ID  SUM(SALARY)
----- -----------
89098     9690.86
89008     6899.89
99308    19999.78

SQL> select tosal(89098) from dual;

TOSAL(89098)
------------
        9691

SQL> select tosal(89008) from dual;

TOSAL(89008)
------------
        6900

SQL> select tosal(99308) from dual;

TOSAL(99308)
------------
       20000

SQL> select tosal(9930) from dual;

TOSAL(9930)
-----------
```

b) Create a function that accepts the factory id and category id, and returns the most expensive raw_material in that category in the factory.

CODE

```
CREATE OR REPLACE FUNCTION maxb
(
  f in factory.fid%type,
  c in category.cid%type
```

```
)
return raw_material.rname%type is
m raw_material.price%type:=0;
mr raw_material.rname%type:=null;
cursor fr is select rname FROM
belongs where fid=f;
frc fr%rowtype;
rn raw_material.rname%type;
cursor pr is select price
from raw_material where
cid=c and rname=rn;
prc pr%rowtype;
BEGIN
for frc in fr loop
rn:=frc.rname;
for prc in pr loop
if m<prc.price THEN
m:=prc.price;
mr:=rn;
end if;
end loop;
end loop;
return mr;
end;
```

```
SQL> CREATE OR REPLACE FUNCTION maxb
  2  (
  3      f in factory.fid%type,
  4      c in category.cid%type
  5  )
  6  return raw_material.rname%type is
  7  m raw_material.price%type:=0;
  8  mr raw_material.rname%type:=null;
  9  cursor fr is select rname FROM
 10  belongs where fid=f;
 11  frc fr%rowtype;
 12  rn raw_material.rname%type;
 13  cursor pr is select price
 14  from raw_material where
 15  cid=c and rname=rn;
 16  prc pr%rowtype;
 17  BEGIN
 18  for frc in fr loop
 19  rn:=frc.rname;
 20  for prc in pr loop
 21  if m<prc.price THEN
 22  m:=prc.price;
 23  mr:=rn;
 24  end if;
 25  end loop;
 26  end loop;
 27  return mr;
 28  end;
 29  /

Function created.

SQL> select maxb(45685,56945) from dual;

MAXB(45685,56945)
-----------------------------------------
WOOD

SQL> select maxb(45685,5695) from dual;
MAXB(45685,5695)
-----------------------------------------


SQL> select rname FROM
  2  belongs where fid=45685;

RNAME
--------------------
COCA
WOOD

SQL> select price,cid from
  2  raw_material where cid=56945
  3  and rname in ('COCA','WOOD');

     PRICE CID
---------- -----
    198.34 56945
   2898.34 56945
```

PROCEDURE CREATION

a) Create a procedure that prints the following for all the stores in the database :-

(i) The number of employees in the store.

(ii) The salary and employee name of each employee.

(iii) The total salary of all the employees in the store. (HINT: use previously created function)

(iv) The name, price, brand name, factory and category name of each product available in the store.

(print appropriate message where ever data missing is found)

DATA

```
SQL> select bname,cname,pname,s_id from product natural join brand natural join category natural join contains;

BNAME                CNAME                PNAME                          S_ID
-------------------- -------------------- ------------------------------ -----
TATA                 Kids                 DAIRY MILK                     89098
GOLDEN HARVEST       Grocery              SAMPAN DAL                     89098
TATA                 Kids                 DAIRY MILK                     89008
GOLDEN HARVEST       Grocery              SAMPAN DAL                     89308
TATA                 Kids                 DAIRY MILK                     89308
RELIANCE             Ladies               WOMEN SLEEPER                  99308
TATA                 Kids                 SPONGY SHOES                   98308
TATA                 Kids                 DAIRY MILK                     98308

8 rows selected.

SQL> select s_id from store minus select s_id from employee;

S_ID
-----
89308
98308

SQL> select s_id,eid from store natural join employee;

S_ID  EID
----- -----
89098 35353
89098 33353
99308 73033
99308 70033
89008 70933
```

DATA ENTRY OF STORE TO CHECK THE CODE FOR EMPTY STORE

CODE

INSERT INTO STORE VALUES

```
(
  '55555',
  'BNK FACTORY',
  '7849567897',
  ADR
  (
    '6-A BLOCK',
    'KANPUR',
    879876
  )
);
```

```
SQL> INSERT INTO STORE VALUES
  2  (
  3      '55555',
  4      'BNK FACTORY',
  5      '7849567897',
  6      ADR
  7      (
  8          '6-A BLOCK',
  9          'KANPUR',
 10          879876
 11      )
 12  );

1 row created.
```

CODE FOR THE PROCEDURE

CREATE OR REPLACE PROCEDURE

```
stinfo is
cursor s is select s_id,sname
FROM store;
id store.s_id%type;
cursor em is select ename,salary
from employee where s_id=id;
```

```
cursor co is select pid FROM
contains where s_id=id;
k co%rowtype;
hg product%rowtype;
f int;
j em%rowtype;
i s%rowtype;
bi product.bid%type;
fi product.fid%type;
ci product.cid%type;
cursor bod is select bname
from brand where bid=bi;
cursor cod is select cname
from category where cid=ci;
cursor fod is select fname
from factory where fid=fi;
fna fod%rowtype;
cna cod%rowtype;
bna bod%rowtype;
BEGIN
for i in s loop
id:=i.s_id;
dbms_output.put_line
(
  chr(10)||'Imformation about '
  ||i.sname||chr(10)
);
select count(eid) into f
from employee where s_id=id;
if f=0 then
dbms_output.put_line
(
  'The store is totally automated'
```

```
  ||chr(10)
);
else
dbms_output.put_line
(
  'Number of employees in the store are: '
  ||f||chr(10)||
  'Employee details are: '||chr(10)
);
for j in em loop
dbms_output.put_line
(
  'Name: '||j.ename||chr(10)||
  'Salary: '||j.salary||chr(10)
);
end loop;
dbms_output.put_line
(
  'Total salary of all employees='
  ||tosal(id)
);
end if;
dbms_output.put_line
(
  chr(10)||'Product details are: '
  ||chr(10)
);
f:=0;
for k in co loop
f:=1;
select * into hg from
product where pid=k.pid;
dbms_output.put_line
```

```
(
  'Name: '||hg.pname||chr(10)||
  'Price: '||hg.price
);
bi:=hg.bid;
fi:=hg.fid;
ci:=hg.cid;
open cod;
open fod;
open bod;
fetch cod into cna;
fetch fod into fna;
fetch bod into bna;
dbms_output.put_line
(
  'Category: '||cna.cname||chr(10)||
  'Brand: '||bna.bname||chr(10)||
  'Factory: '||fna.fname||chr(10)
);
close cod;
close fod;
close bod;
end loop;
if f<>1 THEN
dbms_output.put_line
(
  chr(10)||'The store is empty'
  ||chr(10)
);
end if;
end loop;
end;
```

SCREEN SHOT OF CODE

```
SQL> CREATE OR REPLACE PROCEDURE
  2  stinfo is
  3  cursor s is select s_id,sname
  4  FROM store;
  5  id store.s_id%type;
  6  cursor em is select ename,salary
  7  from employee where s_id=id;
  8  cursor co is select pid FROM
  9  contains where s_id=id;
 10  k co%rowtype;
 11  hg product%rowtype;
 12  f int;
 13  j em%rowtype;
 14  i s%rowtype;
 15  bi product.bid%type;
 16  fi product.fid%type;
 17  ci product.cid%type;
 18  cursor bod is select bname
 19  from brand where bid=bi;
 20  cursor cod is select cname
 21  from category where cid=ci;
 22  cursor fod is select fname
 23  from factory where fid=fi;
 24  fna fod%rowtype;
 25  cna cod%rowtype;
 26  bna bod%rowtype;
 27  BEGIN
 28  for i in s loop
 29  id:=i.s_id;
 30  dbms_output.put_line
 31  (
 32      chr(10)||'Imformation about '
 33      ||i.sname||chr(10)
 34  );
 35  select count(eid) into f
 36  from employee where s_id=id;
 37  if f=0 then
 38  dbms_output.put_line
 39  (
 40      'The store is totally automated'
```

```
 41      ||chr(10)
 42  );
 43  else
 44  dbms_output.put_line
 45  (
 46      'Number of employees in the store are: '
 47      ||f||chr(10)||
 48      'Employee details are: '||chr(10)
 49  );
 50  for j in em loop
 51  dbms_output.put_line
 52  (
 53      'Name: '||j.ename||chr(10)||
 54      'Salary: '||j.salary||chr(10)
 55  );
 56  end loop;
 57  dbms_output.put_line
 58  (
 59      'Total salary of all employees='
 60      ||tosal(id)
 61  );
 62  end if;
 63  dbms_output.put_line
 64  (
 65      chr(10)||'Product details are: '
 66      ||chr(10)
 67  );
 68  f:=0;
 69  for k in co loop
 70  f:=1;
 71  select * into hg from
 72  product where pid=k.pid;
 73  dbms_output.put_line
 74  (
 75      'Name: '||hg.pname||chr(10)||
 76      'Price: '||hg.price
 77  );
 78  bi:=hg.bid;
 79  fi:=hg.fid;
 80  ci:=hg.cid;
 81  open cod;
```

```
 82  open fod;
 83  open bod;
 84  fetch cod into cna;
 85  fetch fod into fna;
 86  fetch bod into bna;
 87  dbms_output.put_line
 88  (
 89      'Category: '||cna.cname||chr(10)||
 90      'Brand: '||bna.bname||chr(10)||
 91      'Factory: '||fna.fname||chr(10)
 92  );
 93  close cod;
 94  close fod;
 95  close bod;
 96  end loop;
 97  if f<>1 THEN
 98  dbms_output.put_line
 99  (
100      chr(10)||'The store is empty'
101      ||chr(10)
102  );
103  end if;
104  end loop;
105  end;
106  /

Procedure created.
```

OUTPUT

```
SQL> EXECUTE STINFO;

Imformation about BNK FACTORY

The store is totally automated


Product details are:


The store is empty


Imformation about BIG BAZAAR

Number of employees in the store are: 2
Employee details are:

Name: RAHUL GUPTA
Salary: 4345.43

Name: ROHIN GUPTA
Salary: 5345.43

Total salary of all employees=9691

Product details are:

Name: DAIRY MILK
Price: 22.55
Category: Kids
Brand: TATA
Factory: CHOCLATE FACTORY

Name: SAMPAN DAL
Price: 1000.34
Category: Grocery
Brand: GOLDEN HARVEST
Factory: DAL FACTORY


Imformation about V-MART
Brand: GOLDEN HARVEST
Factory: DAL FACTORY


Imformation about V-MART

Number of employees in the store are: 1
Employee details are:

Name: PEKU SHARMA
Salary: 6899.89

Total salary of all employees=6900

Product details are:

Name: DAIRY MILK
Price: 22.55
Category: Kids
Brand: TATA
Factory: CHOCLATE FACTORY


Imformation about FOOD MALL

The store is totally automated


Product details are:

Name: SAMPAN DAL
Price: 1000.34
Category: Grocery
Brand: GOLDEN HARVEST
Factory: DAL FACTORY

Name: DAIRY MILK
Price: 22.55
Category: Kids
Brand: TATA
Factory: CHOCLATE FACTORY


Imformation about RELAINCE TRENDS


Imformation about RELAINCE TRENDS

Number of employees in the store are: 2
Employee details are:

Name: NALIN GUPTA
Salary: 9999.89

Name: AKASH GUPTA
Salary: 9999.89

Total salary of all employees=20000

Product details are:

Name: WOMEN SLEEPER
Price: 2000.33
Category: Ladies
Brand: RELIANCE
Factory: CHARITY HOUSE


Imformation about WESTSIDE

The store is totally automated


Product details are:

Name: SPONGY SHOES
Price: 550.44
Category: Kids
Brand: TATA
Factory: CHARITY HOUSE

Name: DAIRY MILK
Price: 22.55
Category: Kids
Brand: TATA
Factory: CHOCLATE FACTORY


PL/SQL procedure successfully completed.
```

b) Create a procedure that prints the following for all the factories in the database :-

(i) The name of the factory along with the owner name and the name of the city in which factory in situated.

(ii) The raw materials (category wise) used in the factory, for all the categories in the table category.

(if not present print appropriate message)

(iii) Name of the most expensive raw material in the factory.

(HINT: use previously created function)

(iv) The name of all the products, produced in the factory.

(v) The warehouse id and phone number of the warehouse.

CODE FOR THE PROCEDURE

```
CREATE OR REPLACE PROCEDURE
factinfo is
cursor f is select *
from factory;
i f%rowtype;
cursor c is select cname,cid
from category;
j c%rowtype;
pl category.cid%type;
lk factory.fid%type;
cursor fb is select rname
from belongs natural join
raw_material where fid=lk
and cid=pl;
cursor pro is select pname
FROM product WHERE fid=lk;
k fb%rowtype;
l pro%rowtype;
wn warehouse.wid%type;
wp warehouse.ph_no%type;
begin
for i in f loop
dbms_output.put_line
(
  chr(10)||'Name-'||i.fname
  ||chr(9)||'Owner Name-'||
  i.owner.name||chr(9)||
  'City-'||i.address.city
  ||chr(10)
);
lk:=i.fid;
for j in c loop
dbms_output.put_line
(
  chr(10)||j.cname
```

```
SQL> CREATE OR REPLACE PROCEDURE
  2  factinfo is
  3  cursor f is select *
  4  from factory;
  5  i f%rowtype;
  6  cursor c is select cname,cid
  7  from category;
  8  j c%rowtype;
  9  pl category.cid%type;
 10  lk factory.fid%type;
 11  cursor fb is select rname
 12  from belongs natural join
 13  raw_material where fid=lk
 14  and cid=pl;
 15  cursor pro is select pname
 16  FROM product WHERE fid=lk;
 17  k fb%rowtype;
 18  l pro%rowtype;
 19  wn warehouse.wid%type;
 20  wp warehouse.ph_no%type;
 21  begin
 22  for i in f loop
 23  dbms_output.put_line
 24  (
 25      chr(10)||'Name-'||i.fname
 26      ||chr(9)||'Owner Name-'||
 27      i.owner.name||chr(9)||
 28      'City-'||i.address.city
 29      ||chr(10)
 30  );
 31  lk:=i.fid;
 32  for j in c loop
 33  dbms_output.put_line
 34  (
 35      chr(10)||j.cname
 36  );
 37  pl:=j.cid;
 38  open fb;
 39  fetch fb into k;
 40  if fb%rowcount=0 THEN
 41  close fb;
```

```
);
pl:=j.cid;
open fb;
fetch fb into k;
if fb%rowcount=0 THEN
close fb;
dbms_output.put_line
(
  'There is no raw material used in '
  ||i.fname||' which belongs to '
  ||j.cname||chr(10)
);
else
close fb;
for k in fb loop
dbms_output.put_line(k.rname);
end loop;
dbms_output.put_line
(
  'The most expensive among above is:
'
  ||maxb(lk,pl)
);
end if;
end loop;
dbms_output.put_line
(
  'Products in the '||i.fname||
  ' are:'
);
for l in pro loop
dbms_output.put_line(l.pname);
end loop;
select wid,ph_no into wn,wp from
warehouse where wid=i.wid;
dbms_output.put_line
(
  chr(10)||'Warehouse ID='||wn||
  chr(10)||'Warehouse phone: '
  ||wp||chr(10)
);
end loop;
end;
```

```
42  dbms_output.put_line
43  (
44      'There is no raw material used in '
45      ||i.fname||' which belongs to '
46      ||j.cname||chr(10)
47  );
48  else
49  close fb;
50  for k in fb loop
51  dbms_output.put_line(k.rname);
52  end loop;
53  dbms_output.put_line
54  (
55     'The most expensive among above is: '
56     ||maxb(lk,pl)
57  );
58  end if;
59  end loop;
60  dbms_output.put_line
61  (
62      'Products in the '||i.fname||
63      ' are:'
64  );
65  for l in pro loop
66  dbms_output.put_line(l.pname);
67  end loop;
68  select wid,ph_no into wn,wp from
69  warehouse where wid=i.wid;
70  dbms_output.put_line
71  (
72      chr(10)||'Warehouse ID='||wn||
73      chr(10)||'Warehouse phone: '
74      ||wp||chr(10)
75  );
76  end loop;
77  end;
78  /

Procedure created.

SQL> execute factinfo;
```

OUTPUT

```
SQL> execute factinfo;

Name-TOWN FACTORY       Owner Name-RAHUL GUPTA  City-KANPUR


ELECTRONICS
There is no raw material used in TOWN FACTORY which belongs to ELECTRONICS


Grocery
COCA
WOOD
The most expensive among above is: WOOD

Ladies
There is no raw material used in TOWN FACTORY which belongs to Ladies


Kids
There is no raw material used in TOWN FACTORY which belongs to Kids

Products in the TOWN FACTORY are:
SUPERFAST COMPUTER

Warehouse ID=84645
Warehouse phone: 9900543380

Name-CHARITY HOUSE      Owner Name-DHAVAL MAVANI        City-MUMBAI


ELECTRONICS
There is no raw material used in CHARITY HOUSE which belongs to ELECTRONICS


Grocery
PULSES
The most expensive among above is: PULSES

Ladies
There is no raw material used in CHARITY HOUSE which belongs to Ladies


Kids
There is no raw material used in CHARITY HOUSE which belongs to Kids

Products in the CHARITY HOUSE are:
SPONGY SHOES
WOMEN SLEEPER

Warehouse ID=24345
Warehouse phone: 9800425380
```

```
Name-DAL FACTORY        Owner Name-YASH KHANDELWAL      City-DURGAPUR


ELECTRONICS
There is no raw material used in DAL FACTORY which belongs to ELECTRONICS


Grocery
PULSES
The most expensive among above is: PULSES

Ladies
There is no raw material used in DAL FACTORY which belongs to Ladies


Kids
There is no raw material used in DAL FACTORY which belongs to Kids

Products in the DAL FACTORY are:
SAMPAN DAL

Warehouse ID=84645
Warehouse phone: 9900543380


Name-CHOCLATE FACTORY   Owner Name-RAHUL KUMAR  City-KOLKATA


ELECTRONICS
There is no raw material used in CHOCLATE FACTORY which belongs to ELECTRONICS


Grocery
COCA
The most expensive among above is: COCA

Ladies
There is no raw material used in CHOCLATE FACTORY which belongs to Ladies


Kids
There is no raw material used in CHOCLATE FACTORY which belongs to Kids

Products in the CHOCLATE FACTORY are:
DAIRY MILK

Warehouse ID=24345
Warehouse phone: 9800425380


PL/SQL procedure successfully completed.

SQL> select fname from factory;

FNAME
------------------------------
TOWN FACTORY
CHARITY HOUSE
DAL FACTORY
CHOCLATE FACTORY

SQL> _
```

IMPLEMENT THE FOLLOWING BUSINESS RULES USING TRIGGER

a)If warehouse and all the factories that share it are in the same city then it can be shared by maximum of 3 cities else it can shared by 2 cities.

b)If the address of the warehouse or the factory is unknown so we cannot have that warehouse for that factory, i.e. we should have data about the address of all the warehouse which are used by any factory as well as the address of all the factories which have a warehouse.

c) If the address of a warehouse is to be deleted so it cannot have any factory using it.

d) If the factory is using a warehouse so we cannot delete it's address.

(It must also work even if not null constraint for wid is dropped from table factory)

NOTE: After the factories are inserted into the database they have the independence of switching between different warehouses as per their wish, I.e. they are not liable to the rule (a) but are liable to rules (b), (c) and (d).

DATA

```
SQL> select wid,w.address.city from warehouse w;

WID   ADDRESS.CITY
----- --------------
24345 LUCKNOW
54345 DELHI
84645 KOLKATA

SQL> select fid,wid,f.address.city from factory f;

FID   WID   ADDRESS.CITY
----- ----- --------------
45685 84645 KANPUR
55645 24345 MUMBAI
95645 84645 DURGAPUR
85645 24345 KOLKATA
```

CODE FOR THE TRIGGERS

TRIGGER ON FACTORY

```
create or replace trigger owners
before INSERT or update on factory
for each row
DECLARE
city varchar2(15);
m int:=1;
n int;
cursor c1 is select address
from factory where wid=:new.wid;
i c1%rowtype;
nn exception;
BEGIN
if :new.wid is NULL then
raise nn;
end if;
select address into i
from warehouse where wid=:new.wid;
if i.address is null
or :new.address is NULL THEN
raise_application_error
```

```
SQL> create or replace trigger owners
  2  before INSERT or update on factory
  3  for each row
  4  DECLARE
  5  city varchar2(15);
  6  m int:=1;
  7  n int;
  8  cursor c1 is select address
  9  from factory where wid=:new.wid;
 10  i c1%rowtype;
 11  nn exception;
 12  BEGIN
 13  if :new.wid is NULL then
 14  raise nn;
 15  end if;
 16  select address into i
 17  from warehouse where wid=:new.wid;
 18  if i.address is null
 19  or :new.address is NULL THEN
 20  raise_application_error
 21  (
 22      -20003,
 23      'Both factory address and warehouse
 24       address should be there'
 25  );
```

```
(
  -20003,
  'Both factory address and warehouse
  address should be there'
);
elsif updating THEN
raise nn;
end if;
select count(fid) into n from
factory where wid=:new.wid;
if n<2 or :new.wid=:old.wid THEN
raise nn;
end if;
select w.address.city into city
from warehouse w where wid=:new.wid;
if city<>:new.address.city THEN
m:=0;
else
for i in cl loop
if city<>i.address.city then
m:=0;
exit;
end if;
end loop;
end if;
if n=2 and m=0 THEN
raise_application_error
(
  -20001,
  'Warehouse can be owned by
  maximum 2 cities if they are
  in different city'
);
elsif n=2 THEN
raise nn;
else
raise_application_error
(
  -20002,
  'Warehouse can be owned by
  maximum 3'
);
end if;
exception
when nn then
n:=0;
end;
```

```
26  elsif updating THEN
27  raise nn;
28  end if;
29  select count(fid) into n from
30  factory where wid=:new.wid;
31  if n<2 or :new.wid=:old.wid THEN
32  raise nn;
33  end if;
34  select w.address.city into city
35  from warehouse w where wid=:new.wid;
36  if city<>:new.address.city THEN
37  m:=0;
38  else
39  for i in cl loop
40  if city<>i.address.city then
41  m:=0;
42  exit;
43  end if;
44  end loop;
45  end if;
46  if n=2 and m=0 THEN
47  raise_application_error
48  (
49      -20001,
50      'Warehouse can be owned by
51       maximum 2 cities if they are
52       in different city'
53  );
54  elsif n=2 THEN
55  raise nn;
56  else
57  raise_application_error
58  (
59      -20002,
60      'Warehouse can be owned by
61       maximum 3'
62  );
63  end if;
64  exception
65  when nn then
66  n:=0;
67  end;
68  /

Trigger created.

SQL>
```

TRIGGER ON WAREHOUSE

```
create or replace trigger wa
before update on warehouse
for each row DECLARE
j warehouse.wid%type;
BEGIN
select distinct wid into j from
factory where wid=:new.wid;
if :new.address is null THEN
raise_application_error
(
  -20009,
  'Cannot delete address of
  warehouse that is owned by
  any factory'
);
end if;
exception
when no_data_found THEN
j:=null;
end;
```

```
SQL> create or replace trigger wa
  2  before update on warehouse
  3  for each row DECLARE
  4  j warehouse.wid%type;
  5  BEGIN
  6  select distinct wid into j from
  7  factory where wid=:new.wid;
  8  if :new.address is null THEN
  9  raise_application_error
 10  (
 11      -20009,
 12      'Cannot delete address of
 13      warehouse that is owned by
 14      any factory'
 15  );
 16  end if;
 17  exception
 18  when no_data_found THEN
 19  j:=null;
 20  end;
 21  /

Trigger created.
```

OUTPUT

ERROR-20009

```
SQL> update warehouse set address=null where wid=24345;
update warehouse set address=null where wid=24345
*
ERROR at line 1:
ORA-20009: Cannot delete address of
warehouse that is owned by
any factory
ORA-06512: at "DVSYS.WA", line 7
ORA-04088: error during execution of trigger 'DVSYS.WA'


SQL> update warehouse set address=null where wid=54345;

1 row updated.
```

ERROR-20001

```
SQL> insert into factory(address,fid,wid,fname) values(adr('98','78',787678),67656,24345,'juj');
insert into factory(address,fid,wid,fname) values(adr('98','78',787678),67656,24345,'juj')
            *
ERROR at line 1:
ORA-20001: Warehouse can be owned by
maximum 2 cities if they are
in different city
ORA-06512: at "DVSYS.OWNERS", line 55
ORA-04088: error during execution of trigger 'DVSYS.OWNERS'
```

ERROR-20003

```
SQL> savepoint a;

Savepoint created.

SQL> insert into warehouse(wid) values('78767');

1 row created.

SQL> insert into warehouse(wid,ph_no) values('98767',9898765676);

1 row created.

SQL> insert into warehouse values('68767',adr('agh','ak',878789),9798765676);

1 row created.

SQL> update factory set address=null where fid=45685;
update factory set address=null where fid=45685
       *
ERROR at line 1:
ORA-20003: Both factory address and warehouse
address should be there
ORA-06512: at "DVSYS.OWNERS", line 17
ORA-04088: error during execution of trigger 'DVSYS.OWNERS'


SQL> insert into factory(fid,wid,address,fname) values('34543','78767',adr('kj','kj',898989),'aj');
insert into factory(fid,wid,address,fname) values('34543','78767',adr('kj','kj',898989),'aj')
            *
ERROR at line 1:
ORA-20003: Both factory address and warehouse
address should be there
ORA-06512: at "DVSYS.OWNERS", line 17
ORA-04088: error during execution of trigger 'DVSYS.OWNERS'


SQL> insert into factory(fid,wid,address,fname) values('34543','98767',adr('kj','kj',898989),'aj');
insert into factory(fid,wid,address,fname) values('34543','98767',adr('kj','kj',898989),'aj')
            *
ERROR at line 1:
ORA-20003: Both factory address and warehouse
address should be there
ORA-06512: at "DVSYS.OWNERS", line 17
ORA-04088: error during execution of trigger 'DVSYS.OWNERS'


SQL> insert into factory(fid,wid,address,fname) values('34543','68767',adr('kj','kj',898989),'aj');

1 row created.

SQL> rollback to a;

Rollback complete.
```

ERROR-20002

```
SQL> savepoint a;

Savepoint created.

SQL> insert into factory(address,fid,wid,fname) values(adr('98','DELHI',787678),67656,54345,'juj');

1 row created.

SQL> rollback to a;

Rollback complete.

SQL> savepoint a;

Savepoint created.

SQL> insert into factory(address,fid,wid,fname) values(adr('98','DELHI',787678),67656,54345,'juj');

1 row created.

SQL> insert into factory(address,fid,wid,fname) values(adr('dsd','DELHI',432456),43656,54345,'rfs');

1 row created.

SQL> insert into factory(address,fid,wid,fname) values(adr('a','MUMBAI',434534),41656,54345,'abc');
insert into factory(address,fid,wid,fname) values(adr('a','MUMBAI',434534),41656,54345,'abc')
            *
ERROR at line 1:
ORA-20001: Warehouse can be owned by
maximum 2 cities if they are
in different city
ORA-06512: at "DVSYS.OWNERS", line 54
ORA-04088: error during execution of trigger 'DVSYS.OWNERS'


SQL> insert into factory(address,fid,wid,fname) values(adr('x','DELHI',456345),41156,54345,'qbc');

1 row created.

SQL> insert into factory(address,fid,wid,fname) values(adr('wx','DELHI',156345),91156,54345,'qqc');
insert into factory(address,fid,wid,fname) values(adr('wx','DELHI',156345),91156,54345,'qqc')
            *
ERROR at line 1:
ORA-20002: Warehouse can be owned by
maximum 3
ORA-06512: at "DVSYS.OWNERS", line 64
ORA-04088: error during execution of trigger 'DVSYS.OWNERS'


SQL> rollback to a;

Rollback complete.

SQL>
```

d) A fully automated store can only have at max 2 products.

e) A normal store can have maximum of 4 products.

f) If an employee starts working for a previously fully automated store then he\she cannot work for a category.

g) Salary of employee entering into a fully automated store should not be less than 5000.

h) If an employee is working for a category (i.e cid not null) so his/her salary cannot be less than 4000.

NOTE: These rules are only applicable for a new entry of employee into the database.

DATA

```
SQL> SELECT PID,S_ID FROM CONTAINS WHERE S_ID IN
  2  (SELECT S_ID FROM STORE MINUS SELECT S_ID FROM EMPLOYEE);

PID   S_ID
----- -----
45432 89308
55332 89308
45334 98308
55332 98308
```

```
SQL> select count(pid),s_id from contains group by s_id;

COUNT(PID) S_ID
---------- -----
         2 89308
         4 89098
         1 89008
         2 98308
         1 99308

SQL> select count(eid),s_id from employee group by s_id;

COUNT(EID) S_ID
---------- -----
         2 89098
         1 89008
         2 99308
```

```
SQL> select * from contains;

PID   S_ID  DATE_ADDE
----- ----- ---------
55332 89098 23-AUG-20
45432 89098 13-NOV-19
55332 89008 25-AUG-20
45432 89308 25-AUG-19
55332 89308 27-AUG-20
45382 99308 07-SEP-18
45334 98308 17-JUL-20
55332 98308 17-SEP-18
45432 89098
55332 89098

10 rows selected.
```

CODE FOR THE TRIGGERS

TRIGGER ON CONTAINS

```
create or replace trigger cpc
before insert or update
of s_id on contains
for each row DECLARE
n int;
k int;
nn exception;
BEGIN
select count(pid) into n
from contains where s_id=:new.s_id;
select count(eid) into k from
employee where s_id=:new.s_id;
if k=0 and n>1 THEN
raise_application_error
(
  -20005,
  'Fully automated store cannot have more than 2 products'
);
end if;
if n<4 then
raise nn;
else
raise_application_error
(
  -20004,
  'Store cannot have more than 4 products'
);
end if;
exception
when nn then
n:=0;
end;
```

```
SQL> create or replace trigger cpc
  2  before insert or update
  3  of s_id on contains
  4  for each row DECLARE
  5  n int;
  6  k int;
  7  nn exception;
  8  BEGIN
  9  select count(pid) into n
 10  from contains where s_id=:new.s_id;
 11  select count(eid) into k from
 12  employee where s_id=:new.s_id;
 13  if k=0 and n>1 THEN
 14  raise_application_error
 15  (
 16      -20005,
 17      'Fully automated store cannot have more than 2 products'
 18  );
 19  end if;
 20  if n<4 then
 21  raise nn;
 22  else
 23  raise_application_error
 24  (
 25      -20004,
 26      'Store cannot have more than 4 products'
 27  );
 28  end if;
 29  exception
 30  when nn then
 31  n:=0;
 32  end;
 33  /

Trigger created.
```

TRIGGER ON EMPLOYEE

create or replace trigger eps

```
before insert  on employee
for each row DECLARE
e store.s_id%type;
a int;
n int;
nn exception;
BEGIN
select count(eid) into a FROM
employee where s_id=:new.s_id;
select count(pid) into n FROM
contains where s_id=:new.s_id;
if :new.cid is not NULL THEN
if a=0 THEN
raise_application_error
(
  -20007,
  'Employee entering into a fully
  automated store cannot work for a
  category'
);
end if;
if :new.salary<4000 THEN
raise_application_error
(
  -20008,
  'Employee working for a category
  cannot have salary less than 4000'
);
end if;
raise nn;
end if;
if a=0 and :new.salary<5000 THEN
raise_application_error
(
  -20010,
  'Employee entering into a fully automated store
  cannot have salary less than 5000'
);
end if;
exception
when nn then
n:=0;
end;
```

```
SQL> create or replace trigger eps
  2  before insert  on employee
  3  for each row DECLARE
  4  e store.s_id%type;
  5  a int;
  6  n int;
  7  nn exception;
  8  BEGIN
  9  select count(eid) into a FROM
 10  employee where s_id=:new.s_id;
 11  select count(pid) into n FROM
 12  contains where s_id=:new.s_id;
 13  if :new.cid is not NULL THEN
 14  if a=0 THEN
 15  raise_application_error
 16  (
 17      -20007,
 18      'Employee entering into a fully
 19      automated store cannot work for a
 20      category'
 21  );
 22  end if;
 23  if :new.salary<4000 THEN
 24  raise_application_error
 25  (
 26      -20008,
 27      'Employee working for a category
 28      cannot have salary less than 4000
 29  );
 30  end if;
 31  raise nn;
 32  end if;
 33  if a=0 and :new.salary<5000 THEN
 34  raise_application_error
 35  (
 36      -20010,
 37      'Employee entering into a fully automated store
 38      cannot have salary less than 5000'
 39  );
 40  end if;
 41  exception
 42  when nn then
 43  n:=0;
 44  end;
 45  /

Trigger created.
```

OUTPUT

ERROR-20004

```
SQL> insert into contains(s_id,pid) values(89098,45382);
insert into contains(s_id,pid) values(89098,45382)
*
ERROR at line 1:
ORA-20004: Store cannot have more than 4 products
ORA-06512: at "DVSYS.CPC", line 20
ORA-04088: error during execution of trigger 'DVSYS.CPC'
```

ERROR-20005

```
SQL> insert into contains(s_id,pid) values(89308,45382);
insert into contains(s_id,pid) values(89308,45382)
            *
ERROR at line 1:
ORA-20005: Fully automated store cannot have more than 2 products
ORA-06512: at "DVSYS.CPC", line 11
ORA-04088: error during execution of trigger 'DVSYS.CPC'
```

ERROR-20007

```
SQL> insert into employee(eid,ename,s_id,cid) values(32454,'f','89308',34945);
insert into employee(eid,ename,s_id,cid) values(32454,'f','89308',34945)
            *
ERROR at line 1:
ORA-20007: Employee entering into a fully
automated store cannot work for a
category
ORA-06512: at "DVSYS.EPS", line 13
ORA-04088: error during execution of trigger 'DVSYS.EPS'
```

ERROR-20008

```
SQL> insert into employee(eid,ename,s_id,cid,salary) values(32454,'f','89008',34945,3000);
insert into employee(eid,ename,s_id,cid,salary) values(32454,'f','89008',34945,3000)
           *
ERROR at line 1:
ORA-20008: Employee working for a category
cannot have salary less than 4000
ORA-06512: at "DVSYS.EPS", line 22
ORA-04088: error during execution of trigger 'DVSYS.EPS'
```

ERROR-20010

```
SQL> insert into employee(eid,ename,s_id,salary) values(32454,'f','89308',4032);
insert into employee(eid,ename,s_id,salary) values(32454,'f','89308',4032)
            *
ERROR at line 1:
ORA-20010: Employee entering into a fully automated store
cannot have salary less than 5000
ORA-06512: at "DVSYS.EPS", line 32
ORA-04088: error during execution of trigger 'DVSYS.EPS'
```