**Hamna B Mustafa**

**Hbm170002**

|  |  |
| --- | --- |
| **Semester Project – Part 2** | **Database Design – Fall 2021** |
| **Due Date: check eLearning** | **Intrinsic detail of this project is discussed during lecture** |

There is an MM Company, which purchases some parts from vendors to produce some products. It has several departments, marketing sites, and parts supply vendors in the company.

1. For each department, department id and department name will be recorded.
2. People in the company can be divided into three types -- employees, customers, and potential employees. Each person can belong to more than one type. Each person in the company has the following attributes: Personal\_ID, Name (Last Name, First Name), Age (below 65), Gender, Address (address line 1, address line 2, city, state, zipcode), and Phone number (one individual may have more than one phone number). For customers, his/her preferred salesmen were recorded in the system. For employees, Rank and Title (e.g. CEO, Principle, Partner, etc.) will be recorded for them.
3. Each employee of the company must have only one direct supervisor, while each supervisor can have several supervisees. One employee can work for one or more departments at different time. But at one time, one employee can only work for one department. The system needs to record start time and end time for each shift among different department for one employee.
4. Each job position’s information is recorded to hire new people. It contains the Job ID, job description, and posted date in the system.
5. The job positions are posted by the departments. Both existing employees and potential employees can apply each job post by any department. The company will select some candidates from the applications and make interviews.
6. For each job position, several interviews will be made to select a suitable person.
7. For each interview, candidates (interviewees), interviewers, job position and interview time are recorded. After each round interview, the interviewers give a grade to it ranging from 0 to 100. The grade over 60 represents that the interviewee pass the interview. One person is selected when her/his average grade is over 70 and she/he passes at least 5 rounds of interviews.

8) For each product in the company, the system needs to record Product ID, Product Type, Size, List Price, Weight, and Style.

9) There are many marketing sites for the company. For each site, Site ID, Site Name, and Site Location are recorded.

10) There are several people working for each site, and meanwhile, one person can work on different sites. It is able to track the details of each sale history --- salesmen, customers, product, sale time, and sites.

11) Part purchase is also a vital activity in the company. The system needs to record each vendor’s Vendor ID, Name, Address, Account Number, Credit Rating, and Purchasing Web Service URL.

12) One vendor may supply many types of parts. The price of the same part type may vary from different vendors but the price of one part type of one vendor will keep same. It is able to track which part types used in each product and the number of each type of part used for the product.

13) In addition, the system maintains the information of each employee’s monthly salary which includes transaction number, pay\_date, and amount (Note: transaction number could be same among different employees. However, for each employee, the transaction number is **UNIQUE**).

**Project Description**

In this project, we are given requirements by a company to create a database for them. Using those requirements, we create a conceptual model – an EER diagram. We then translate that EER diagram into a relational schema. Once we have constructed our initial relational schema, we check to make sure it normalized to 3NF and do so if it isn’t. Then, we fix our EER diagram if it needs fixing.

After we have a normalized updated relational schema, we convert that into an actual database using SQL and Oracle. We use CREATE statements to create all our tables. Then we fill the database with some dummy data so that we can perform the given queries and create some views.

**Project Questions**

1. Can you think 5 more rules (other than the one explicitly described above) that are likely to be used in a company.
   1. An interview time for a job cannot be prior to the date the job was posted
   2. Candidates must be at least 18 years old
   3. Every department must have at least 1 employee
   4. Each employee must work at least one site
   5. Every vendor must have a minimum credit rating of 700
2. Is the ability to model super-class/subclass relationships likely to be important in such environment? Why or why not?  
     
   This ability is very important because most databases like these will have some sort of inheritance relationships. It is very important to be able to share some attributes and not others, especially if there are many differing attributes. The ability to model super-class/subclass relationships does not only reduce redundancy and increase efficiency but it also makes the representation of many different environments possible.
3. Justify using a Relational DBMS like Oracle for this project.  
     
   In this project, the usage of a relational DBMS was incredibly important. A relational database can store different tables that are related to one another. It can connect the tables that are related to each other without increasing redundancy. This project had many different parts that were all heavily related to one another. Thus, it was crucial to have a system that would be able to record such relations.

**Project Exercises**

1. Draw an EER to accurately represent this set of requirements. This will be your Conceptual Design. Clearly specify any assumption that you are making. You can use any tools (software) to draw the EER. You don’t need describe the value constraints of the attributions in the EER diagram. (25%)

Assumptions:

1. Assuming that Department\_id and Department\_name are unique
2. Assuming that Personal\_id is unique
3. Assuming that department can have one or more employees
4. Assuming that Job\_Id is unique
5. Assuming that site\_id is unique
6. Assuming that product\_id is unique
7. Assuming that part\_type is unique
8. Assuming that vendor\_id is unique
9. Assuming that each interview is done by 1 interviewer
10. Assuming that once a candidate is selected from the interviews, they are immediately hired

Diagram

Description automatically generated

**II.** Use a relational DBMS to implement the database. Perform the following steps. (20%)

1. Convert your Conceptual model to a Logical model that can be implemented in a relational DBMS like Oracle. During this process you replace M-N relationships and multi-valued attributes with constructs that can be implemented in the relational DBMS. Draw EER for the logical model after your modifications. Feel free to change your conceptual model (first delivery) if needed.

Diagram

Description automatically generated with medium confidence

**b)** Convert the EER to a database design. Document your design in Database Schema format like the one we discussed in the class.

**III.** Use appropriate naming conventions for all your tables and attributes. (40%)

1. Normalize all your tables to third normal form. Make any necessary changes to the EER. Explain why these changes needed to be made.

My relational schema is automatically in 1NF due to the very definition of a flat relational model. Checking to see if all the tables are in 2NF and 3NF:

**PERSON:**The PERSON table only has 1 attribute in its PRIMARY KEY. Hence, the 2NF test need not be applied as it is automatically 2NF. Furthermore, there are no transitive dependencies of any nonkey attributes on the PRIMARY KEY. So it is also in 3NF.

**PHONE\_NUMBERS:**

The PHONE\_NUMBERS table doesn’t have any nonprime attributes. Hence, the 2NF test need not be applied as it is automatically 2NF. Because of the same reason, it is also automatically in 3NF.

**SALARY:**Assuming the following functional dependencies exist:

{Personal\_id, Transaction\_no} 🡪 Pay\_date, Amount

Both the nonprime attributes (Pay\_date and Amount) are fully functionally dependent on the **PRIMARY KEY** so the table is in 2NF.

There are no transitive dependencies of any nonkey attributes on the PRIMARY KEY. So it is also in 3NF.

**WORKS\_FOR:**

Assuming the following functional dependencies exist:

{Personal\_id, D\_id} 🡪 Start\_time, End\_time

Both the nonprime attributes (Start\_time, End\_time) are fully functionally dependent on the **PRIMARY KEY** so the table is in 2NF.  
There are no transitive dependencies of any nonkey attributes on the PRIMARY KEY. So it is also in 3NF.

**DEPARTMENT:**

The DEPARTMENT table only has 1 attribute in its PRIMARY KEY. Hence, the 2NF test need not be applied as it is automatically 2NF.  
There are no transitive dependencies of any nonkey attributes on the PRIMARY KEY. So it is also in 3NF.

**JOB:**

Assuming the following functional dependencies exist:  
{D\_id, Job\_id} 🡪 Job\_date, Description

Both the nonprime attributes (Job\_date, Description) are fully functionally dependent on the PRIMARY KEY so the table is in 2NF.

There are no transitive dependencies of any nonkey attributes on the PRIMARY KEY. So it is also in 3NF.

**APPLIES\_TO:**The APPLIES\_TO table doesn’t have any nonprime attributes. Hence, the 2NF test need not be applied as it is automatically 2NF. For the same reason, it also automatically in 3NF.

**INTERVIEW:**

Assuming the following functional dependencies exist:  
{Candidate\_id, Employee\_id, Job\_id, Round} 🡪 Grade, Interview\_time

{Candidate\_id} 🡪 Candidate\_email

Both the nonprime attributes (Grade, Interview\_time) are fully functionally dependent on the PRIMARY KEY. However, Candidate\_email is only partially dependent on the PRIMARY KEY so the table is not in 2NF. To fix that, we will split the table to make it 2NF:  
  
**INTERVIEW**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Candidate\_id | Employee\_id | Job\_id | Round | Grade | Interview\_time |

**INTERVIEW\_EMAIL**

|  |  |
| --- | --- |
| C**andidate\_id** | Candidate\_email |

There are no transitive dependencies of any nonkey attributes on the PRIMARY KEY. So it is in 3NF.

**SITE:**  
The SITE table only has 1 attribute in its PRIMARY KEY. Hence, the 2NF test need not be applied as it is automatically 2NF.  
There are no transitive dependencies of any nonkey attributes on the PRIMARY KEY. So it is also in 3NF.

**SITE\_EMPLOYEE:**The SITE\_EMPLOYEE table doesn’t have any nonprime attributes. Hence, the 2NF test need not be applied as it is automatically 2NF. For the same reason, it also automatically in 3NF.

**PRODUCT:**

The PRODUCT table only has 1 attribute in its PRIMARY KEY. Hence, the 2NF test need not be applied as it is automatically 2NF.  
There are no transitive dependencies of any nonkey attributes on the PRIMARY KEY. So it is also in 3NF.

**SALE:**

The SALE table doesn’t have any nonprime attributes. Hence, the 2NF test need not be applied as it is automatically 2NF. For the same reason, it also automatically in 3NF.

**PART:**

The PART table only has 1 attribute in its PRIMARY KEY. Hence, the 2NF test need not be applied as it is automatically 2NF.  
There are no transitive dependencies of any nonkey attributes on the PRIMARY KEY. So it is also in 3NF.

**SUPPLIED\_BY:**

Assuming the following functional dependencies exist:  
{Vendor\_id, Part\_type} 🡪 Price, Weight

The nonprime attributes (Price, Weight) are fully functionally dependent on the PRIMARY KEY so the table is in 2NF.  
There are no transitive dependencies of any nonkey attributes on the PRIMARY KEY. So it is also in 3NF.

**VENDOR:**  
The VENDOR table only has 1 attribute in its PRIMARY KEY. Hence, the 2NF test need not be applied as it is automatically 2NF.  
There are no transitive dependencies of any nonkey attributes on the PRIMARY KEY. So it is also in 3NF.

**PART\_PRODUCT\_VENDOR**:  
The PART\_PRODUCT\_VENDOR table doesn’t have any nonprime attributes. Hence, the 2NF test need not be applied as it is automatically 2NF. For the same reason, it also automatically in 3NF.

The following relational schema is in 1NF, 2NF and 3NF;

Diagram

Description automatically generated

The following is the updated EER after removing M-N relationships and multivalued attributes as well as normalizing:

Diagram

Description automatically generated

1. Draw a dependency diagram for each table

The relational schema shown above contains the dependency diagram.

**c)** Write SQL statements to create database, tables, and all other structures. PRIMARY KEYs and FOREIGN KEYs must be defined appropriately. The quantity constraints of the relation between the entities, which should be described in EER diagram, are not required.

**CREATE TABLE** PERSON

(

Personal\_id VARCHAR2(9) **NOT NULL**,

Lname VARCHAR2(10) **NOT NULL**,

Fname VARCHAR2(10) **NOT NULL**,

Gender VARCHAR2(1),

Age NUMBER,

Address\_1 VARCHAR2(30),

Address\_2 VARCHAR2(30),

City VARCHAR2(10),

State VARCHAR2(10),

Zipcode NUMBER,

Employee\_flag NUMBER(1),

Super\_id VARCHAR2(9),

Title VARCHAR2(30),

Rank VARCHAR2(30),

Potential\_flag NUMBER(1),

Customer\_flag NUMBER(1),

Pref\_salesmen VARCHAR2(20),

Candidate\_flag NUMBER(1),

**PRIMARY KEY** (Personal\_id),

**FOREIGN KEY**(Super\_id) **REFERENCES** PERSON(Personal\_id));

**CREATE TABLE** PHONE\_NUMBERS

(

Personal\_id VARCHAR2(9) **NOT NULL**,

Phone\_no VARCHAR2(10) **NOT NULL**,

**PRIMARY KEY** (Personal\_id, Phone\_no),

**FOREIGN KEY**(Personal\_id) **REFERENCES** PERSON(Personal\_id));

**CREATE TABLE** SALARY

(

Personal\_id VARCHAR2(9) **NOT NULL**,

Transaction\_no NUMBER **NOT NULL**,

Pay\_date DATE,

Amount NUMBER,

**PRIMARY KEY**(Personal\_id, Transaction\_no),

**FOREIGN KEY**(Personal\_id) **REFERENCES** PERSON(Personal\_id));

**CREATE TABLE** DEPARTMENT

(

Dep\_id NUMBER **NOT NULL**,

D\_name VARCHAR(20) **NOT NULL**,

**PRIMARY KEY**(Dep\_id),

**UNIQUE**(D\_name));

**CREATE TABLE** WORKS\_FOR

(

Personal\_id VARCHAR2(9) **NOT NULL**,

Dep\_id NUMBER **NOT NULL**,

Start\_time TIMESTAMP **NOT NULL**,

End\_time TIMESTAMP,

**PRIMARY KEY**(Personal\_id, Dep\_id, Start\_time),

**FOREIGN KEY**(Personal\_id) **REFERENCES** PERSON(Personal\_id),

**FOREIGN KEY**(Dep\_id) **REFERENCES** DEPARTMENT(Dep\_id));

**CREATE TABLE** JOB

(

Job\_id NUMBER **NOT NULL**,

Dep\_id NUMBER **NOT NULL**,

Job\_date DATE,

Description VARCHAR(150),

**PRIMARY KEY**(Job\_id),

**FOREIGN KEY**(Dep\_id) **REFERENCES** DEPARTMENT(Dep\_id));

**CREATE TABLE** APPLIES\_TO

(

Candidate\_id VARCHAR(9) **NOT NULL**,

Job\_id NUMBER **NOT NULL**,

**PRIMARY KEY**(Candidate\_id, Job\_id),

**FOREIGN KEY**(Candidate\_id) **REFERENCES** PERSON(Personal\_id));

**CREATE TABLE** INTERVIEW

(

Candidate\_id VARCHAR(9) **NOT NULL**,

Employee\_id VARCHAR(9) **NOT NULL**,

Job\_id NUMBER **NOT NULL**,

Round NUMBER **NOT NULL**,

Grade NUMBER,

Interview\_time TIMESTAMP,

**PRIMARY KEY**(Candidate\_id, Employee\_id, Job\_id, Round),

**FOREIGN KEY**(Candidate\_id) **REFERENCES** PERSON(Personal\_id),

**FOREIGN KEY**(Employee\_id) **REFERENCES** PERSON(Personal\_id),

**FOREIGN KEY**(Job\_id) **REFERENCES** JOB(Job\_id));

**CREATE TABLE** INTERVIEW\_EMAIL

(

Candidate\_id VARCHAR2(9) **NOT NULL**,

Email VARCHAR2(50),

**PRIMARY KEY**(Candidate\_id),

**FOREIGN KEY**(Candidate\_id) **REFERENCES** PERSON(Personal\_id));

**CREATE TABLE** SITE

(

Site\_id NUMBER **NOT NULL**,

Site\_name VARCHAR(20) **NOT NULL**,

Site\_location VARCHAR(50),

**PRIMARY KEY**(Site\_id));

**CREATE TABLE** SITE\_EMPLOYEE

(

Employee\_id VARCHAR(9) **NOT NULL**,

Site\_id NUMBER **NOT NULL**,

**PRIMARY KEY**(Employee\_id, Site\_id),

**FOREIGN KEY**(Employee\_id) **REFERENCES** PERSON(Personal\_id),

**FOREIGN KEY**(Site\_id) **REFERENCES** SITE(Site\_id));

**CREATE TABLE** PRODUCT

(

Product\_id NUMBER **NOT NULL**,

Product\_type VARCHAR2(20) **NOT NULL**,

List\_price NUMBER,

Product\_size NUMBER,

Weight NUMBER,

Product\_style VARCHAR2(20),

**PRIMARY KEY**(Product\_id),

**UNIQUE**(Product\_type));

**CREATE TABLE** SALE

(

Customer\_id VARCHAR2(9) **NOT NULL**,

Employee\_id VARCHAR2(9) **NOT NULL**,

Product\_id NUMBER **NOT NULL**,

Site\_id NUMBER **NOT NULL**,

Sale\_time TIMESTAMP **NOT NULL**,

**PRIMARY KEY**(Customer\_id, Employee\_id, Product\_id, Site\_id, Sale\_time),

**FOREIGN KEY**(Customer\_id) **REFERENCES** PERSON(Personal\_id),

**FOREIGN KEY**(Employee\_id) **REFERENCES** PERSON(Personal\_id),

**FOREIGN KEY**(Product\_id) **REFERENCES** PRODUCT(Product\_id),

**FOREIGN KEY**(Site\_id) **REFERENCES** SITE(Site\_id));

**CREATE TABLE** PART

(

Part\_type VARCHAR(20) **NOT NULL**,

Product\_id NUMBER,

Part\_amount NUMBER,

**PRIMARY KEY**(Part\_type),

**FOREIGN KEY**(Product\_id) **REFERENCES** PRODUCT(Product\_id));

**CREATE TABLE** VENDOR

(

Vendor\_id NUMBER **NOT NULL**,

Vendor\_name VARCHAR2(20) **NOT NULL**,

Vendor\_address VARCHAR2(50),

Account\_no NUMBER,

Credit\_rating NUMBER,

Vendor\_Url VARCHAR(50),

**PRIMARY KEY**(Vendor\_id));

**CREATE TABLE** SUPPLIED\_BY

(

Vendor\_id NUMBER **NOT NULL**,

Part\_type VARCHAR(20) **NOT NULL**,

Price NUMBER,

Weight NUMBER,

**PRIMARY KEY**(Vendor\_id, Part\_type),

**FOREIGN KEY**(Vendor\_id) **REFERENCES** VENDOR(Vendor\_id),

**FOREIGN KEY**(Part\_type) **REFERENCES** PART(Part\_type));

**CREATE TABLE** PART\_PRODUCT\_VENDOR

(

Part\_type VARCHAR(20) **NOT NULL**,

Product\_id NUMBER **NOT NULL**,

Vendor\_id NUMBER **NOT NULL**,

**PRIMARY KEY**(Part\_type, product\_id, vendor\_id),

**FOREIGN KEY**(Part\_type) **REFERENCES** PART(Part\_type),

**FOREIGN KEY**(Product\_id) **REFERENCES** PRODUCT(Product\_id),

**FOREIGN KEY**(Vendor\_id) **REFERENCES** VENDOR(Vendor\_id));

**d)** Use the Create View statement to create the following views:

**1)** View1: This view returns the average salary each employee has earned from

the company monthly after she/he becomes an employee in the company.

CREATE VIEW AVG\_SALARIES(Lastname, Employee\_id, avgsalary)

AS SELECT lname, SALARY.personal\_id, AVG(amount)

FROM SALARY, PERSON

WHERE SALARY.personal\_id = PERSON.personal\_id

GROUP BY SALARY.personal\_id, lname;

SELECT \*

FROM AVG\_SALARIES;

Table

Description automatically generated

**2)** View2: This view returns the number of interviews rounds each interviewee pass for each job position.

CREATE VIEW NO\_ROUNDS\_PASSED(candidate\_id,Lastname,No\_of\_rounds\_passed)

AS SELECT personal\_id, lname, COUNT(\*)

FROM PERSON, INTERVIEW

WHERE candidate\_flag=1 AND personal\_id = candidate\_id AND grade >= 60

GROUP BY personal\_id, lname;

SELECT \*

FROM NO\_ROUNDS\_PASSED;

Table

Description automatically generated

**3)** View3: This view returns the number of items of each product type sold.

CREATE VIEW NO\_OF\_ITEMS\_SOLD(Product\_id, product\_type, items\_sold)

AS SELECT PRODUCT.product\_id, product\_type, COUNT(\*)

FROM PRODUCT, SALE

WHERE PRODUCT.product\_id = SALE.product\_id

GROUP BY PRODUCT.product\_id, product\_type;

SELECT \*

FROM NO\_OF\_ITEMS\_SOLD;

Table

Description automatically generated

**4)** View4: This view returns the part purchase cost for each product.

CREATE VIEW COST\_OF\_PRODUCT(Product\_id, Purchase\_cost)

AS SELECT PART\_PRODUCT\_VENDOR.product\_id, SUM(price\*part\_amount)

FROM PART\_PRODUCT\_VENDOR, SUPPLIED\_BY, PART

WHERE PART\_PRODUCT\_VENDOR.Vendor\_id = SUPPLIED\_BY.vendor\_id AND PART\_PRODUCT\_VENDOR.part\_type = SUPPLIED\_BY.part\_type AND PART\_PRODUCT\_VENDOR.part\_type = PART.part\_type

GROUP BY PART\_PRODUCT\_VENDOR.product\_id;

SELECT \*

FROM COST\_OF\_PRODUCT;

Table

Description automatically generated with medium confidence

**e)** Answer the following Queries. Feel free to use any of the views that you created in part (d).

1) Return the ID and Name of interviewers who participate in interviews where the interviewee’s name is “Hellen Cole” arranged for job “11111”.

SELECT DISTINCT personal\_id, Fname, Lname

FROM PERSON, INTERVIEW

WHERE candidate\_id = (SELECT personal\_id

FROM PERSON

WHERE Lname = 'Cole' AND Fname = 'Hellen') AND employee\_id = personal\_id AND job\_id = 11111;

Table

Description automatically generated

2) Return the ID of all jobs which are posted by department “Marketing” in January 2011.

SELECT job\_id, job\_date

FROM JOB, DEPARTMENT

WHERE JOB.dep\_id = DEPARTMENT.dep\_id AND d\_name = 'Marketing' AND to\_char(job\_date,'YYYY') = 2011 AND to\_char(job\_date,'MM') = 01;

A picture containing calendar

Description automatically generated

3) Return the ID and Name of the employees having no supervisees.

SELECT fname, lname, personal\_id

FROM PERSON

WHERE personal\_id NOT IN (SELECT super\_id

FROM PERSON

WHERE employee\_flag = 1 AND super\_id IS NOT NULL) AND employee\_flag =1;

Table

Description automatically generated

4) Return the Id and Location of the marketing sites which have no sale records during March, 2011.

SELECT site\_id, site\_location

FROM SITE

WHERE NOT EXISTS ( SELECT site\_id

FROM SALE  
WHERE to\_char(sale\_time, 'MM') = 03 AND to\_char(sale\_time, 'YYYY') = 2011 AND SALE.site\_id = SITE.site\_id);

Table

Description automatically generated with medium confidence

5) Return the job’s id and description which does not hire a suitable person one month after it is posted.

SELECT job\_id, description

FROM JOB

WHERE job\_id NOT IN (SELECT INTERVIEW.job\_id

FROM INTERVIEW, JOB

WHERE candidate\_id IN (SELECT candidate\_id

FROM INTERVIEW

GROUP BY candidate\_id

HAVING count(\*) >= 5 AND AVG(grade) >= 70) AND round = 5 AND JOB.job\_id = INTERVIEW.job\_id AND interview\_time <= ADD\_MONTHS(job\_date,1)) ;

Text, table

Description automatically generated

6) Return the ID and Name of the salesmen who have sold all product type whose price is above $200.

SELECT DISTINCT Fname, lname, personal\_id

FROM PERSON, SALE, PRODUCT

WHERE Employee\_id = personal\_id AND SALE.product\_id = PRODUCT.product\_id AND list\_price >= 200;

Table

Description automatically generated

7) Return the department’s id and name which has no job post during 1/1/2011 and 2/1/2011.

SELECT dep\_id, d\_name

FROM DEPARTMENT

WHERE NOT EXISTS (SELECT dep\_id

FROM JOB

WHERE JOB.dep\_id = DEPARTMENT.dep\_id AND (to\_char(job\_date, 'YYYY-MM-DD') = '2011-01-01' OR to\_char(job\_date, 'YYYY-MM-DD') = '2011-02-01'));

Table

Description automatically generated

8) Return the ID, Name, and Department ID of the existing employees who apply job “12345”.

SELECT DISTINCT PERSON.personal\_id, Fname, lname, WORKS\_FOR.dep\_id

FROM PERSON, WORKS\_FOR, APPLIES\_TO

WHERE PERSON.personal\_id = candidate\_id AND job\_id = 12345 AND PERSON.personal\_id = WORKS\_FOR.personal\_id;

Table

Description automatically generated

9) Return the best seller’s type in the company (sold the most items).

SELECT product\_Type, COUNT(Product\_type)

FROM PRODUCT, SALE

WHERE PRODUCT.product\_id = SALE.product\_id

GROUP BY Product\_Type

HAVING COUNT(Product\_Type)=(SELECT MAX(mycount)

FROM (SELECT product\_type, COUNT(\*) mycount

FROM PRODUCT, SALE

WHERE PRODUCT.product\_id = SALE.product\_id

GROUP BY product\_type));

Table

Description automatically generated with low confidence

10) Return the product type whose net profit is highest in the company (money earned minus the part cost).

SELECT PRODUCT.Product\_id, product\_type, (PRODUCT.list\_price - SUM(price\*part\_amount)) as net\_profit

FROM PART\_PRODUCT\_VENDOR, SUPPLIED\_BY, PART, PRODUCT

WHERE PART\_PRODUCT\_VENDOR.Vendor\_id = SUPPLIED\_BY.vendor\_id AND PART\_PRODUCT\_VENDOR.part\_type = SUPPLIED\_BY.part\_type AND PART\_PRODUCT\_VENDOR.part\_type = PART.part\_type AND PART\_PRODUCT\_VENDOR.product\_id = PRODUCT.product\_id

GROUP BY PRODUCT.Product\_id, Product\_type, list\_price

HAVING PRODUCT.list\_price - SUM(price\*part\_amount) = (SELECT MAX(profit)

FROM (SELECT PART\_PRODUCT\_VENDOR.product\_id, (PRODUCT.list\_price - SUM(price\*part\_amount)) profit

FROM PART\_PRODUCT\_VENDOR, SUPPLIED\_BY, PART, PRODUCT

WHERE PART\_PRODUCT\_VENDOR.Vendor\_id = SUPPLIED\_BY.vendor\_id AND PART\_PRODUCT\_VENDOR.part\_type = SUPPLIED\_BY.part\_type AND PART\_PRODUCT\_VENDOR.part\_type = PART.part\_type AND PART\_PRODUCT\_VENDOR.product\_id = PRODUCT.product\_id

GROUP BY PART\_PRODUCT\_VENDOR.product\_id, list\_price));

A picture containing shape

Description automatically generated

11) Return the name and id of the employees who has worked in all departments after hired by the company.

SELECT PERSON.personal\_id, Fname, lname

FROM PERSON, WORKS\_FOR

WHERE PERSON.personal\_id = WORKS\_FOR.personal\_id

GROUP BY PERSON.personal\_id, fname, lname

HAVING COUNT(\*) = (SELECT count(\*)

FROM DEPARTMENT);

Table

Description automatically generated

12) Return the name and email address of the interviewee who is selected.  
  
SELECT INTERVIEW.candidate\_id, Fname, Lname, email

FROM INTERVIEW , JOB, INTERVIEW\_EMAIL, PERSON

WHERE INTERVIEW.candidate\_id IN (SELECT candidate\_id

FROM INTERVIEW

WHERE round = 5 AND Grade >= 60) AND JOB.job\_id = INTERVIEW.job\_id AND INTERVIEW.candidate\_id = INTERVIEW\_EMAIL.candidate\_id AND INTERVIEW.candidate\_id = personal\_id

GROUP BY INTERVIEW.candidate\_id, Fname, Lname, email

HAVING AVG(GRADE) >= 70;

Table

Description automatically generated

13) Retrieve the name, phone number, email address of the interviewees elected for all the jobs they apply.

SELECT DISTINCT Fname,Lname, phone\_no, email

FROM INTERVIEW, PERSON, PHONE\_NUMBERS, INTERVIEW\_EMAIL

WHERE INTERVIEW.candidate\_id NOT IN (SELECT candidate\_id

FROM INTERVIEW

GROUP BY candidate\_id, job\_id

HAVING count(\*) < 5 AND AVG(grade) < 70)

AND INTERVIEW.candidate\_id IN (SELECT candidate\_id

FROM INTERVIEW

GROUP BY candidate\_id, job\_id

HAVING count(\*) >= 5 AND AVG(grade) >= 70)

AND INTERVIEW.candidate\_id = PERSON.personal\_id AND INTERVIEW.candidate\_id = PHONE\_NUMBERS.personal\_id AND INTERVIEW.candidate\_id = INTERVIEW\_EMAIL.candidate\_id;

Table

Description automatically generated with medium confidence

14) Return the employee’s name and id whose average monthly salary is highest in the company.  
  
SELECT Fname, Lname, PERSON.Personal\_id, AVG(amount)

FROM PERSON, SALARY

WHERE PERSON.Personal\_id = SALARY.Personal\_id

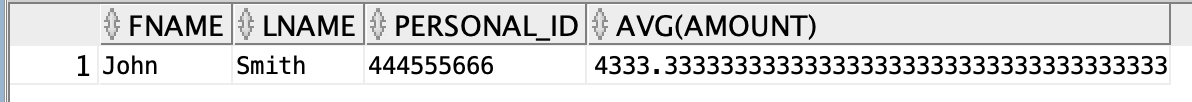
GROUP BY Fname, Lname, PERSON.Personal\_id

HAVING AVG(amount) = (SELECT MAX(avgsalary)

FROM (SELECT Personal\_id, AVG(amount) avgsalary

FROM SALARY

GROUP BY Personal\_id));



15) Return the ID and Name of the vendor who supply part whose name is “Cup” and weight is smaller than 4 pound and the price is lowest among all vendors.

SELECT VENDOR.Vendor\_id, vendor\_name, price

FROM SUPPLIED\_BY, VENDOR

WHERE price = (SELECT MIN(price)

FROM SUPPLIED\_BY

WHERE Part\_type = 'Cup' AND Weight < 4) AND VENDOR.vendor\_id = SUPPLIED\_BY.vendor\_id;

A picture containing text

Description automatically generated

**IV.** Document the final term project report. (15%)

**a)** Problem description

**b)** Project questions (Answer questions listed in this project).

**c)** EER diagram with all assumptions.

**d)** Relation schema after normalization. All relations must be in 3NF. The relation schema should include PRIMARY KEYs as well as FOREIGN KEYs (if any) for all relations.

**e)** All requested SQL statements.

**f)** Dependency diagram.

The following are my SQL statements that I used to insert dummy data:

INSERT ALL

INTO PERSON (Personal\_id, Lname, Fname, Gender, Age, Address\_1, Address\_2, City, State, Zipcode, Employee\_flag, Super\_id, Title, Rank, Potential\_flag, Customer\_flag, Pref\_salesmen, Candidate\_flag) VALUES ('123456789','Mustafa','Hamna','F',22, '123 Lane', null, 'Dallas', 'TX', 77043, 1, null, 'Developer', 'Junior',0,0,null,1)

INTO PERSON (Personal\_id, Lname, Fname, Gender, Age, Address\_1, Address\_2, City, State, Zipcode, Employee\_flag, Super\_id, Title, Rank, Potential\_flag, Customer\_flag, Pref\_salesmen, Candidate\_flag) VALUES ('111222333','Cole','Hellen','F',25, 'Apt 23', '456 Lane', 'Richardson', 'TX', 77043, 0, null, null, null,1,0,null,1)

INTO PERSON (Personal\_id, Lname, Fname, Gender, Age, Address\_1, Address\_2, City, State, Zipcode, Employee\_flag, Super\_id, Title, Rank, Potential\_flag, Customer\_flag, Pref\_salesmen, Candidate\_flag) VALUES ('444555666','Smith','John','M',30, '789 Lane', null, 'Dallas', 'TX', 77074, 1, 123456789, 'Assistant', 'Junior',0,0,null,0)

INTO PERSON (Personal\_id, Lname, Fname, Gender, Age, Address\_1, Address\_2, City, State, Zipcode, Employee\_flag, Super\_id, Title, Rank, Potential\_flag, Customer\_flag, Pref\_salesmen, Candidate\_flag) VALUES ('777888999','Granger','Hermione','M',40, 'Chestnut Lane', null, 'Addison', 'TX', 75134, 1, null, 'Saleswoman', 'Senior',0,0,null,0)

INTO PERSON (Personal\_id, Lname, Fname, Gender, Age, Address\_1, Address\_2, City, State, Zipcode, Employee\_flag, Super\_id, Title, Rank, Potential\_flag, Customer\_flag, Pref\_salesmen, Candidate\_flag) VALUES ('333222111','Potter','Harry','M',42, '4 Privet Drive', null, 'Plano', 'TX', 77123, 0, null, null, null,1,0,null,1)

INTO PERSON (Personal\_id, Lname, Fname, Gender, Age, Address\_1, Address\_2, City, State, Zipcode, Employee\_flag, Super\_id, Title, Rank, Potential\_flag, Customer\_flag, Pref\_salesmen, Candidate\_flag) VALUES ('999888777','Weasley','Ron','M',43, 'Apt 11', '111 St', 'Dallas', 'TX', 71111, 0, null, null, null,0,1,'Hermione Granger',0)

INTO PERSON (Personal\_id, Lname, Fname, Gender, Age, Address\_1, Address\_2, City, State, Zipcode, Employee\_flag, Super\_id, Title, Rank, Potential\_flag, Customer\_flag, Pref\_salesmen, Candidate\_flag) VALUES ('101112134','Holmes','Sherlock','M',56, '221B', 'Baker St', 'Richardson', 'TX', 75074,1, '777888999','Salesman', 'Junior',0,0,null,1)

INTO PERSON (Personal\_id, Lname, Fname, Gender, Age, Address\_1, Address\_2, City, State, Zipcode, Employee\_flag, Super\_id, Title, Rank, Potential\_flag, Customer\_flag, Pref\_salesmen, Candidate\_flag) VALUES ('111111111','Lovegood','Luna','F',20, '111 Walnut Ln', null, 'Richardson', 'TX', 75074,0, null,null, null,0,1,'Hamna Mustafa',0)

INTO PERSON (Personal\_id, Lname, Fname, Gender, Age, Address\_1, Address\_2, City, State, Zipcode, Employee\_flag, Super\_id, Title, Rank, Potential\_flag, Customer\_flag, Pref\_salesmen, Candidate\_flag) VALUES ('222222222','Weasley','Ginny','F',17, '111 Walnut Ln', null, 'Richardson', 'TX', 75074,0, null,null, null,0,1,'Hamna Mustafa',0)

INTO PERSON (Personal\_id, Lname, Fname, Gender, Age, Address\_1, Address\_2, City, State, Zipcode, Employee\_flag, Super\_id, Title, Rank, Potential\_flag, Customer\_flag, Pref\_salesmen, Candidate\_flag) VALUES ('333333333','Bing','Chandler','M',50, '987 Portside Ln', null, 'Richardson', 'TX', 75074,0, null,null, null,0,1,'Hamna Mustafa',0)

select \* from dual;

INSERT ALL

INTO PHONE\_NUMBERS(Personal\_id,Phone\_no) VALUES ('111222333','4691234567')

INTO PHONE\_NUMBERS(Personal\_id,Phone\_no) VALUES ('111222333','2261234567')

INTO PHONE\_NUMBERS(Personal\_id,Phone\_no) VALUES ('101112134','4691011123')

INTO PHONE\_NUMBERS(Personal\_id,Phone\_no) VALUES ('123456789','4691112223')

INTO PHONE\_NUMBERS(Personal\_id,Phone\_no) VALUES ('333222111','7131015433')

INTO PHONE\_NUMBERS(Personal\_id,Phone\_no) VALUES ('444555666','9876543210')

INTO PHONE\_NUMBERS(Personal\_id,Phone\_no) VALUES ('777888999','1111234567')

INTO PHONE\_NUMBERS(Personal\_id,Phone\_no) VALUES ('999888777','7138882225')

INTO PHONE\_NUMBERS(Personal\_id,Phone\_no) VALUES ('999888777','2263853957')

INTO PHONE\_NUMBERS(Personal\_id,Phone\_no) VALUES ('123456789','4295729572')

select \* from dual;

INSERT ALL

INTO SALARY(personal\_id,transaction\_no,pay\_date,amount) VALUES (123456789, 1, to\_date('2011-01-01','yyyy,mm,dd'),2000)

INTO SALARY(personal\_id,transaction\_no,pay\_date,amount) VALUES (123456789, 2, to\_date('2011-02-01','yyyy,mm,dd'),2100)

INTO SALARY(personal\_id,transaction\_no,pay\_date,amount) VALUES (123456789, 3, to\_date('2011-03-01','yyyy,mm,dd'),2000)

INTO SALARY(personal\_id,transaction\_no,pay\_date,amount) VALUES (123456789, 4, to\_date('2011-04-01','yyyy,mm,dd'),3200)

INTO SALARY(personal\_id,transaction\_no,pay\_date,amount) VALUES (123456789, 5, to\_date('2011-05-01','yyyy,mm,dd'),2000)

INTO SALARY(personal\_id,transaction\_no,pay\_date,amount) VALUES (123456789, 6, to\_date('2011-06-01','yyyy,mm,dd'),2000)

INTO SALARY(personal\_id,transaction\_no,pay\_date,amount) VALUES (444555666, 1, to\_date('2011-01-01','yyyy,mm,dd'),4000)

INTO SALARY(personal\_id,transaction\_no,pay\_date,amount) VALUES (444555666, 2, to\_date('2011-02-01','yyyy,mm,dd'),5000)

INTO SALARY(personal\_id,transaction\_no,pay\_date,amount) VALUES (444555666, 3, to\_date('2011-03-01','yyyy,mm,dd'),3500)

INTO SALARY(personal\_id,transaction\_no,pay\_date,amount) VALUES (444555666, 4, to\_date('2011-04-01','yyyy,mm,dd'),4000)

INTO SALARY(personal\_id,transaction\_no,pay\_date,amount) VALUES (444555666, 5, to\_date('2011-05-01','yyyy,mm,dd'),5000)

INTO SALARY(personal\_id,transaction\_no,pay\_date,amount) VALUES (444555666, 6, to\_date('2011-06-01','yyyy,mm,dd'),4500)

INTO SALARY(personal\_id,transaction\_no,pay\_date,amount) VALUES (777888999, 1, to\_date('2011-01-01','yyyy,mm,dd'),500)

INTO SALARY(personal\_id,transaction\_no,pay\_date,amount) VALUES (777888999, 2, to\_date('2011-02-01','yyyy,mm,dd'),2000)

INTO SALARY(personal\_id,transaction\_no,pay\_date,amount) VALUES (777888999, 3, to\_date('2011-03-01','yyyy,mm,dd'),700)

INTO SALARY(personal\_id,transaction\_no,pay\_date,amount) VALUES (777888999, 4, to\_date('2011-04-01','yyyy,mm,dd'),1000)

INTO SALARY(personal\_id,transaction\_no,pay\_date,amount) VALUES (777888999, 5, to\_date('2011-05-01','yyyy,mm,dd'),1500)

INTO SALARY(personal\_id,transaction\_no,pay\_date,amount) VALUES (777888999, 6, to\_date('2011-06-01','yyyy,mm,dd'),500)

INTO SALARY(personal\_id,transaction\_no,pay\_date,amount) VALUES (101112134, 1, to\_date('2011-01-01','yyyy,mm,dd'),2000)

INTO SALARY(personal\_id,transaction\_no,pay\_date,amount) VALUES (101112134, 2, to\_date('2011-02-01','yyyy,mm,dd'),3000)

INTO SALARY(personal\_id,transaction\_no,pay\_date,amount) VALUES (101112134, 3, to\_date('2011-03-01','yyyy,mm,dd'),3000)

INTO SALARY(personal\_id,transaction\_no,pay\_date,amount) VALUES (101112134, 4, to\_date('2011-04-01','yyyy,mm,dd'),3000)

INTO SALARY(personal\_id,transaction\_no,pay\_date,amount) VALUES (101112134, 5, to\_date('2011-05-01','yyyy,mm,dd'),3000)

INTO SALARY(personal\_id,transaction\_no,pay\_date,amount) VALUES (101112134, 6, to\_date('2011-06-01','yyyy,mm,dd'),3000)

select \* from dual;

INSERT ALL

INTO DEPARTMENT(Dep\_id,D\_name) VALUES (1,'Finance')

INTO DEPARTMENT(Dep\_id,D\_name) VALUES (2,'Marketing')

INTO DEPARTMENT(Dep\_id,D\_name) VALUES (3,'HR')

INTO DEPARTMENT(Dep\_id,D\_name) VALUES (4,'Development')

INTO DEPARTMENT(Dep\_id,D\_name) VALUES (5,'Sales')

select \* from dual;

INSERT ALL

INTO WORKS\_FOR(Personal\_id,Dep\_id,Start\_time,End\_time) VALUES (123456789,4, timestamp '2011-01-01 09:00:00',timestamp '2011-01-01 17:00:00')

INTO WORKS\_FOR(Personal\_id,Dep\_id,Start\_time,End\_time) VALUES (123456789,4, timestamp'2011-03-01 09:00:00',timestamp '2011-03-01 17:00:00')

INTO WORKS\_FOR(Personal\_id,Dep\_id,Start\_time,End\_time) VALUES (444555666,1, timestamp '2011-02-01 09:00:00',timestamp '2011-02-01 17:00:00')

INTO WORKS\_FOR(Personal\_id,Dep\_id,Start\_time,End\_time) VALUES (777888999,5, timestamp '2011-01-15 09:00:00',timestamp '2011-01-15 17:00:00')

INTO WORKS\_FOR(Personal\_id,Dep\_id,Start\_time,End\_time) VALUES (101112134,5, timestamp '2011-06-01 09:00:00',timestamp '2011-06-01 17:00:00')

INTO WORKS\_FOR(personal\_id, dep\_id, start\_time, end\_time) VALUES (444555666, 2, timestamp '2011-02-02 09:00:00',timestamp '2011-02-02 17:00:00')

INTO WORKS\_FOR(personal\_id, dep\_id, start\_time, end\_time) VALUES (444555666, 3, timestamp '2011-02-02 09:00:00',timestamp '2011-02-02 17:00:00')

INTO WORKS\_FOR(personal\_id, dep\_id, start\_time, end\_time) VALUES (444555666, 4, timestamp '2011-02-03 09:00:00',timestamp '2011-02-02 17:00:00')

INTO WORKS\_FOR(personal\_id, dep\_id, start\_time, end\_time) VALUES (444555666, 5, timestamp '2011-02-04 09:00:00',timestamp '2011-02-02 17:00:00')

select \* from dual;

INSERT ALL

INTO JOB(Dep\_id,Job\_id,Job\_date,Description) VALUES (2,11111, to\_date('2011-01-01','yyyy,mm,dd'),'Marketing job folks!')

INTO JOB(Dep\_id,Job\_id,Job\_date,Description) VALUES (2,22222, to\_date('2011-01-23','yyyy,mm,dd'),'Another Marketing job folks!')

INTO JOB(Dep\_id,Job\_id,Job\_date,Description) VALUES (1,33333, to\_date('2011-03-15','yyyy,mm,dd'),'Finance job folks!')

INTO JOB(Dep\_id,Job\_id,Job\_date,Description) VALUES (1,44444, to\_date('2011-04-05','yyyy,mm,dd'),'Another finance job folks!')

INTO JOB(Dep\_id,Job\_id,Job\_date,Description) VALUES (3,12345, to\_date('2011-05-01','yyyy,mm,dd'),'HR job folks!')

INTO JOB(Dep\_id,Job\_id,Job\_date,Description) VALUES (3,55555, to\_date('2011-02-25','yyyy,mm,dd'),'Another HR job folks!')

INTO JOB(Dep\_id,Job\_id,Job\_date,Description) VALUES (4,66666, to\_date('2011-01-01','yyyy,mm,dd'),'Development job folks!')

INTO JOB(Dep\_id,Job\_id,Job\_date,Description) VALUES (5,77777, to\_date('2011-03-01','yyyy,mm,dd'),'Sales job folks!')

select \* from dual;

INSERT ALL

INTO APPLIES\_TO(candidate\_id,job\_id) VALUES (111222333,11111)

INTO APPLIES\_TO(candidate\_id,job\_id) VALUES (111222333,22222)

INTO APPLIES\_TO(candidate\_id,job\_id) VALUES (123456789,12345)

INTO APPLIES\_TO(candidate\_id,job\_id) VALUES (101112134,12345)

INTO APPLIES\_TO(candidate\_id,job\_id) VALUES (333222111,12345)

select \* from dual;

INSERT ALL

INTO INTERVIEW(candidate\_id, employee\_id,job\_id,round,grade,interview\_time) VALUES (111222333, 444555666, 11111, 1, 80, timestamp '2011-01-15 12:30:00')

INTO INTERVIEW(candidate\_id, employee\_id,job\_id,round,grade,interview\_time) VALUES (111222333, 101112134, 11111, 2, 70, timestamp '2011-01-17 13:30:00')

INTO INTERVIEW(candidate\_id, employee\_id,job\_id,round,grade,interview\_time) VALUES (111222333, 101112134, 11111, 3, 80, timestamp '2011-01-20 12:00:00')

INTO INTERVIEW(candidate\_id, employee\_id,job\_id,round,grade,interview\_time) VALUES (111222333, 777888999, 11111, 4, 80, timestamp '2011-01-25 14:00:00')

INTO INTERVIEW(candidate\_id, employee\_id,job\_id,round,grade,interview\_time) VALUES (111222333, 777888999, 11111, 5, 80, timestamp '2011-01-29 15:30:00')

INTO INTERVIEW(candidate\_id, employee\_id,job\_id,round,grade,interview\_time) VALUES (123456789, 101112134, 12345, 1, 90, timestamp '2011-06-01 12:30:00')

INTO INTERVIEW(candidate\_id, employee\_id,job\_id,round,grade,interview\_time) VALUES (123456789, 101112134, 12345, 2, 90, timestamp '2011-06-05 10:00:00')

INTO INTERVIEW(candidate\_id, employee\_id,job\_id,round,grade,interview\_time) VALUES (123456789, 777888999, 12345, 3, 95, timestamp '2011-06-10 10:30:00')

INTO INTERVIEW(candidate\_id, employee\_id,job\_id,round,grade,interview\_time) VALUES (123456789, 777888999, 12345, 4, 85, timestamp '2011-06-15 11:00:00')

INTO INTERVIEW(candidate\_id, employee\_id,job\_id,round,grade,interview\_time) VALUES (123456789, 777888999, 12345, 5, 90, timestamp '2011-06-20 11:30:00')

INTO INTERVIEW(candidate\_id, employee\_id,job\_id,round,grade,interview\_time) VALUES (333222111, 123456789, 12345, 1, 40, timestamp '2011-06-01 15:00:00')

INTO INTERVIEW(candidate\_id, employee\_id,job\_id,round,grade,interview\_time) VALUES (101112134, 777888999, 12345, 1, 70, timestamp '2011-06-01 09:00:00')

INTO INTERVIEW(candidate\_id, employee\_id,job\_id,round,grade,interview\_time) VALUES (101112134, 777888999, 12345, 2, 50, timestamp '2011-06-01 09:30:00')

INTO INTERVIEW(candidate\_id, employee\_id,job\_id,round,grade,interview\_time) VALUES (123456789, 777888999, 11111, 1, 50, timestamp '2011-06-01 09:30:00')

select \* from dual;

INSERT ALL

INTO INTERVIEW\_EMAIL(candidate\_id, email) VALUES (111222333, 'cole.hellen@gmail.com')

INTO INTERVIEW\_EMAIL(candidate\_id, email) VALUES (123456789, 'hamna.mustafa@gmail.com')

INTO INTERVIEW\_EMAIL(candidate\_id, email) VALUES (333222111, 'the\_chosen\_one@gmail.com')

INTO INTERVIEW\_EMAIL(candidate\_id, email) VALUES (101112134, 'sherlock.holmes@gmail.com')

select \* from dual;

INSERT ALL

INTO SITE(Site\_id,Site\_name,Site\_location) VALUES (1,'Headquarters','111 Special Lane, Dallas TX 77043')

INTO SITE(Site\_id,Site\_name,Site\_location) VALUES (2,'Plano Branch','222 Special Lane, Plano TX 75555')

INTO SITE(Site\_id,Site\_name,Site\_location) VALUES (3,'Richardson Branch','333 Special Lane, Richardson TX 73333')

select \* from dual;

INSERT ALL

INTO SITE\_EMPLOYEE(Employee\_id, Site\_id) VALUES(123456789,1)

INTO SITE\_EMPLOYEE(Employee\_id, Site\_id) VALUES(444555666,2)

INTO SITE\_EMPLOYEE(Employee\_id, Site\_id) VALUES(444555666,3)

INTO SITE\_EMPLOYEE(Employee\_id, Site\_id) VALUES(777888999,1)

INTO SITE\_EMPLOYEE(Employee\_id, Site\_id) VALUES(777888999,2)

INTO SITE\_EMPLOYEE(Employee\_id, Site\_id) VALUES(777888999,3)

INTO SITE\_EMPLOYEE(Employee\_id, Site\_id) VALUES(101112134,1)

INTO SITE\_EMPLOYEE(Employee\_id, Site\_id) VALUES(101112134,2)

INTO SITE\_EMPLOYEE(Employee\_id, Site\_id) VALUES(101112134,3)

select \* from dual;

INSERT ALL

INTO PRODUCT(Product\_id,Product\_type,list\_price,product\_size,weight,product\_style) VALUES (1, 'Cellphone', 1000, 10,7,'Blue color')

INTO PRODUCT(Product\_id,Product\_type,list\_price,product\_size,weight,product\_style) VALUES (2, 'Cup set', 100, 10,3,'White and circular')

INTO PRODUCT(Product\_id,Product\_type,list\_price,product\_size,weight,product\_style) VALUES (3, 'Clips', 20, 10,7,'Multicolor')

INTO PRODUCT(Product\_id,Product\_type,list\_price,product\_size,weight,product\_style) VALUES (4, 'Chair', 300, 10,50,'Black and sleek')

INTO PRODUCT(Product\_id,Product\_type,list\_price,product\_size,weight,product\_style) VALUES (5, 'Candle', 25, 10,3,'Vanilla Scented')

select \* from dual;

INSERT ALL

INTO SALE(Customer\_id,Employee\_id,Product\_id,Site\_id,Sale\_time) VALUES (999888777,101112134,1,1, timestamp '2011-06-20 12:15:00')

INTO SALE(Customer\_id,Employee\_id,Product\_id,Site\_id,Sale\_time) VALUES (999888777,101112134,1,1, timestamp '2011-05-20 14:32:17')

INTO SALE(Customer\_id,Employee\_id,Product\_id,Site\_id,Sale\_time) VALUES (999888777,777888999,1,1, timestamp '2011-04-20 9:07:00')

INTO SALE(Customer\_id,Employee\_id,Product\_id,Site\_id,Sale\_time) VALUES (999888777,777888999,1,1, timestamp '2011-02-20 15:54:21')

INTO SALE(Customer\_id,Employee\_id,Product\_id,Site\_id,Sale\_time) VALUES (999888777,777888999,1,1, timestamp '2011-01-20 12:20:00')

INTO SALE(Customer\_id,Employee\_id,Product\_id,Site\_id,Sale\_time) VALUES (999888777,777888999,1,1, timestamp '2011-06-25 16:15:35')

INTO SALE(Customer\_id,Employee\_id,Product\_id,Site\_id,Sale\_time) VALUES (111111111,123456789,3,2, timestamp '2011-03-25 10:45:00')

INTO SALE(Customer\_id,Employee\_id,Product\_id,Site\_id,Sale\_time) VALUES (111111111,777888999,5,2, timestamp '2011-03-25 10:50:00')

INTO SALE(Customer\_id,Employee\_id,Product\_id,Site\_id,Sale\_time) VALUES (222222222,444555666,2,3, timestamp '2011-04-20 11:23:33')

INTO SALE(Customer\_id,Employee\_id,Product\_id,Site\_id,Sale\_time) VALUES (333333333,123456789,4,3, timestamp '2011-01-05 12:45:00')

INTO SALE(Customer\_id,Employee\_id,Product\_id,Site\_id,Sale\_time) VALUES (333333333,123456789,1,1, timestamp '2011-01-05 16:58:33')

select \* from dual;

INSERT ALL

INTO PART(Part\_type, Product\_id, Part\_amount) VALUES ('Cup', 2, 12)

INTO PART(Part\_type, Product\_id, Part\_amount) VALUES ('Exterior', 1, 1)

INTO PART(Part\_type, Product\_id, Part\_amount) VALUES ('Interior', 1, 1)

INTO PART(Part\_type, Product\_id, Part\_amount) VALUES ('Chip', 1, 2)

INTO PART(Part\_type, Product\_id, Part\_amount) VALUES ('Camera', 1, 3)

INTO PART(Part\_type, Product\_id, Part\_amount) VALUES ('Clip', 3, 6)

INTO PART(Part\_type, Product\_id, Part\_amount) VALUES ('Chair leg', 4, 4)

INTO PART(Part\_type, Product\_id, Part\_amount) VALUES ('Chair arm', 4, 2)

INTO PART(Part\_type, Product\_id, Part\_amount) VALUES ('Cushion', 4, 2)

INTO PART(Part\_type, Product\_id, Part\_amount) VALUES ('Wax', 5, 1)

INTO PART(Part\_type, Product\_id, Part\_amount) VALUES ('Wick', 5, 3)

INTO PART(Part\_type, Product\_id, Part\_amount) VALUES ('Jar', 5, 1)

select \* from dual;

INSERT ALL

INTO VENDOR(vendor\_id,vendor\_name,vendor\_address,account\_no,credit\_rating,vendor\_url) VALUES (1,'GAGA CO', '321 Gaga Lane, Houston, TX 77043',123456,750,'www.gagaco.com')

INTO VENDOR(vendor\_id,vendor\_name,vendor\_address,account\_no,credit\_rating,vendor\_url) VALUES (2,'LALA CO', '321 Lala Lane, Houston, TX 77043',123789,800,'www.lalaco.com')

INTO VENDOR(vendor\_id,vendor\_name,vendor\_address,account\_no,credit\_rating,vendor\_url) VALUES (3,'DADA CO', '321 Dada Lane, Houston, TX 77043',456789,700,'www.dadaco.com')

select \* from dual;

INSERT ALL

INTO SUPPLIED\_BY(vendor\_id,part\_type,price,weight) VALUES (1, 'Cup', 10,3)

INTO SUPPLIED\_BY(vendor\_id,part\_type,price,weight) VALUES (2, 'Cup', 5,2)

INTO SUPPLIED\_BY(vendor\_id,part\_type,price,weight) VALUES (3, 'Cup', 15,3)

INTO SUPPLIED\_BY(vendor\_id,part\_type,price,weight) VALUES (3, 'Exterior', 200,10)

INTO SUPPLIED\_BY(vendor\_id,part\_type,price,weight) VALUES (3, 'Interior', 50,3)

INTO SUPPLIED\_BY(vendor\_id,part\_type,price,weight) VALUES (3, 'Chip', 100,1)

INTO SUPPLIED\_BY(vendor\_id,part\_type,price,weight) VALUES (3, 'Camera', 100,4)

INTO SUPPLIED\_BY(vendor\_id,part\_type,price,weight) VALUES (2, 'Clip', 1,1)

INTO SUPPLIED\_BY(vendor\_id,part\_type,price,weight) VALUES (1, 'Clip', 2,1)

INTO SUPPLIED\_BY(vendor\_id,part\_type,price,weight) VALUES (3, 'Clip', 3,1)

INTO SUPPLIED\_BY(vendor\_id,part\_type,price,weight) VALUES (1, 'Chair leg', 10,10)

INTO SUPPLIED\_BY(vendor\_id,part\_type,price,weight) VALUES (1, 'Chair arm', 10,10)

INTO SUPPLIED\_BY(vendor\_id,part\_type,price,weight) VALUES (1, 'Cushion', 30,5)

INTO SUPPLIED\_BY(vendor\_id,part\_type,price,weight) VALUES (2, 'Cushion', 25,6)

INTO SUPPLIED\_BY(vendor\_id,part\_type,price,weight) VALUES (2, 'Wax', 3,0.5)

INTO SUPPLIED\_BY(vendor\_id,part\_type,price,weight) VALUES (2, 'Wick', 1,0.1)

INTO SUPPLIED\_BY(vendor\_id,part\_type,price,weight) VALUES (2, 'Jar', 2,1)

INTO SUPPLIED\_BY(vendor\_id,part\_type,price,weight) VALUES (3, 'Jar', 3,2)

INTO SUPPLIED\_BY(vendor\_id,part\_type,price,weight) VALUES (1, 'Jar', 1,1)

select \* from dual;

INSERT ALL

INTO PART\_PRODUCT\_VENDOR(Part\_type, product\_id, vendor\_id) VALUES ('Cup',2,2)

INTO PART\_PRODUCT\_VENDOR(Part\_type, product\_id, vendor\_id) VALUES ('Exterior',1,3)

INTO PART\_PRODUCT\_VENDOR(Part\_type, product\_id, vendor\_id) VALUES ('Interior',1,3)

INTO PART\_PRODUCT\_VENDOR(Part\_type, product\_id, vendor\_id) VALUES ('Chip',1,3)

INTO PART\_PRODUCT\_VENDOR(Part\_type, product\_id, vendor\_id) VALUES ('Camera',1,3)

INTO PART\_PRODUCT\_VENDOR(Part\_type, product\_id, vendor\_id) VALUES ('Clip',3,2)

INTO PART\_PRODUCT\_VENDOR(Part\_type, product\_id, vendor\_id) VALUES ('Chair leg',4,1)

INTO PART\_PRODUCT\_VENDOR(Part\_type, product\_id, vendor\_id) VALUES ('Chair arm',4,1)

INTO PART\_PRODUCT\_VENDOR(Part\_type, product\_id, vendor\_id) VALUES ('Cushion',4,1)

INTO PART\_PRODUCT\_VENDOR(Part\_type, product\_id, vendor\_id) VALUES ('Wax',5,2)

INTO PART\_PRODUCT\_VENDOR(Part\_type, product\_id, vendor\_id) VALUES ('Wick',5,2)

INTO PART\_PRODUCT\_VENDOR(Part\_type, product\_id, vendor\_id) VALUES ('Jar',5,1)

select \* from dual;