**Put assumptions/concerns/clarifications per requirement bullet in the document:**

1. For each department, department id and department name will be recorded.

*Assumptions: N/A*

*Concerns: N/A*

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.

*Assumptions: Rank and Title are interchangeable (only one attribute)*

*Concerns: Constraint for age, multivalued attributes phone and address*

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.

*Assumptions: Not all departments have a supervisor (some employees may not have a direct supervisor) and not all supervisors have supervisees*

*Concerns: Have to keep track of active department*

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.

*Assumptions: Job\_id same structure as personal\_id*

*Concerns: N/A*

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.

*Assumptions: If a current employee applies for a job, they are guaranteed to have an interview*

*Concerns: have to make a subclass to lump together potential and current employees into applicants*

6) For each job position, several interviews will be made to select a suitable person.

*Assumptions: doesn’t have to be a certain number - variable*

*Concerns: have to keep track of interview num*

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.

*Assumptions: The first person to pass all 5 interviews for a position gets the job, not everyone gets 5 interviews, each interview is conducted by one person to one other person*

*Concerns: Have to average grade in sql*

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

*Assumptions: Product id is similar to other id’s in structure, type will be the name and style will be the material it’s made of*

*Concerns: style and type are ambiguous*

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

*Assumptions: Site name will be marketing*

*Concerns: N/A*

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.

*Assumptions: each individual sale is completed by a single salesman to a single customer on a single site*

*Concerns: the site location will be attached to sale, not person, since people can work on multiple sites and sites have multiple people*

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.

*Assumptions: this address will not be multivalued, these are not ‘people’ in the company*

*Concerns: N/A*

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.

*Assumptions:* *no more than 9999 number of parts of one type, a product does not use the same type of part from different vendors*

*Concerns: will need to make an entity for parts*

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).

*Assumptions: the salaries can vary over the months, so they’re each potentially different amount values*

*Concerns: not connected to shift so it’s not related to time worked*

PERSON



|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Fname | Lname | Person\_id | Age | Sex | Line\_one | Line\_two | City | State | Zip |

PHONE\_NUM

|  |  |
| --- | --- |
| Person\_id | Phone\_num |



CUSTOMER

|  |  |
| --- | --- |
| Person\_id | Pref\_salesman |

POTENTIAL\_EMPLOYEE

|  |
| --- |
| Person\_id |

EMPLOYEE

|  |  |  |  |
| --- | --- | --- | --- |
| Person\_id | Dept\_id | Rank | Curr\_dept |



APPLICANT

|  |  |
| --- | --- |
| Person\_id | Email |



DEPARTMENT

|  |  |
| --- | --- |
| Department\_id | Department\_name |

SHIFT

|  |  |  |  |
| --- | --- | --- | --- |
| Person\_id | Dept\_id | Time\_begin | Time\_end |



JOB

|  |  |  |  |
| --- | --- | --- | --- |
| Job\_id | Dept\_id | Description | Posted\_date |



INTERVIEW

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Job\_id | Person\_id | Interview\_num | Interviewer\_id | Interview\_grade | Interview\_time |



MONTHLY\_SALARY

|  |  |  |  |
| --- | --- | --- | --- |
| Transaction\_num | Person\_id | Pay\_date | Amount |



SUPERVISOR

|  |  |
| --- | --- |
| Person\_id | Dept\_id |

SITE

|  |  |  |
| --- | --- | --- |
| Site\_id | Location | Site\_name |

PRODUCT

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Product\_id | Product\_type | Size | Style | Weight | Product\_price |

SELLS

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Product\_id | Time | Salesman\_id | Customer\_id | Site\_id |

PARTS

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Product\_id | Vendor\_id | Part\_type | Part\_cost | Num\_parts | Weight |

VENDOR

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Vendor\_id | Vendor\_URL | Vendor\_acc\_num | Credit\_rating | Vendor\_addr | Vendor\_name |

PERSON

|  |
| --- |
| Person\_id |

**Physical Schema:**

**PERSON**

Fname VARCHAR(30)

Lname VARCHAR(30)

Person\_id NUMBER(5)

Age NUMBER(2)

Sex CHAR(1)

Line\_one VARCHAR(30)

Line\_two VARCHAR(30)

City VARCHAR(30)

State CHAR(2)

Zip NUMBER(5)

**PHONE\_NUM**

Person\_id NUMBER(5)

Phone\_num NUMBER(10)

**CUSTOMER**

Person\_id NUMBER(5)

Pref\_salesman VARCHAR(30)

**POTENTIAL\_EMPLOYEE**

Person\_id NUMBER(5)

**EMPLOYEE**

Person\_id NUMBER(5)

Dept\_id NUMBER(5)

Rank VARCHAR(30)

Curr\_dept BOOLEAN

**APPLICANT**

Person\_id NUMBER(5)

Email VARCHAR(30)

**DEPARTMENT**

Department\_id NUMBER(5)

Department\_name VARCHAR(30)

**SHIFT**

Person\_id NUMBER(5)

Dept\_id NUMBER(5)

Time\_begin TIMESTAMP

Time\_end TIMESTAMP

**JOB**

Job\_id NUMBER(5)

Dept\_id NUMBER(5)

Description VARCHAR(150)

Posted\_date DATE

**INTERVIEW**

Job\_id NUMBER(5)

Person\_id NUMBER(5)

Interview\_num NUMBER(1)

Interviewer\_id NUMBER(5)

Interview\_grade CHAR(1)

Interview\_time TIME

**MONTHLY\_SALARY**

Transaction\_num NUMBER(15)

Person\_id NUMBER(5)

Pay\_date DATE

Amount DECIMAL(10,2)

**SUPERVISOR**

Person\_id NUMBER(5)

Dept\_id NUMBER(5)

**SITE**

Site\_id NUMBER(5)

Location VARCHAR(30)

Site\_name VARCHAR(30)

**PRODUCT**

Product\_id NUMBER(5)

Product\_type VARCHAR(30)

Size DECIMAL(10,2)

Style VARCHAR(30)

Weight DECIMAL(10,2)

Product\_price DECIMAL(10,2)

**SELLS**

Product\_id NUMBER(5)

Time TIME

Salesman\_id NUMBER(5)

Customer\_id NUMBER(5)

Site\_id NUMBER(5)

**PARTS\_ FROM**

Product\_id NUMBER(5)

Vendor\_id NUMBER(5)

Part\_type VARCHAR(30)

Part\_cost DECIMAL(10,2)

Num\_parts NUMBER(4)

Weight DECIMAL(10,2)

**VENDOR**

Vendor\_id NUMBER(5)

Vendor\_URL VARCHAR(30)

Vendor\_acc\_num NUMBER(15)

Credit\_rating NUMBER(15)

Vendor\_addr VARCHAR(30)

Vendor\_name VARCHAR(30)

**Normalization**

- all of the relations are in 1NF : there are no multivalued or nested attributes

Relations with primary keys having multiple attributes:

* PHONE\_NUMBER (all attributes are primary – in 2NF)
* INTERVIEW (no non-key attributes are functionally are dependent on a part of the primary key – in 2NF)
* MONTHLY\_SALARY (no non-key attributes are functionally are dependent on a part of the primary key – in 2NF)
* SELLS (all attributes are primary – in 2NF)
* PARTS\_FROM (no non-key attributes are functionally are dependent on a part of the primary key – in 2NF)

3NF:

Modifications made to:

PARTS\_FROM – added COST\_PARTS and NUM\_PARTS

EMPLOYEE - added CURR\_DEPT

**Dependency Diagrams:**

PERSON

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Fname | Lname | Person\_id | Age | Sex | Line\_one | Line\_two | City | State | Zip |

PHONE\_NUM

|  |  |
| --- | --- |
| Person\_id | Phone\_num |

CUSTOMER

|  |  |
| --- | --- |
| Person\_id | Pref\_salesman |

POTENTIAL\_EMPLOYEE

|  |
| --- |
| Person\_id |

EMPLOYEE

|  |  |  |
| --- | --- | --- |
| Person\_id | Dept\_id | Rank |

APPLICANT

|  |  |
| --- | --- |
| Person\_id | Email |

DEPARTMENT

|  |  |
| --- | --- |
| Department\_id | Department\_name |

CURR\_DEPT

|  |  |  |
| --- | --- | --- |
| Person\_id | Dept\_id | Curr\_dept |

SHIFT

|  |  |  |  |
| --- | --- | --- | --- |
| Person\_id | Dept\_id | Time\_begin | Time\_end |

JOB

|  |  |  |  |
| --- | --- | --- | --- |
| Job\_id | Dept\_id | Description | Posted\_date |

INTERVIEW

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Job\_id | Person\_id | Interview\_num | Interviewer\_id | Interview\_grade | Interview\_time |

MONTHLY\_SALARY

|  |  |  |  |
| --- | --- | --- | --- |
| Transaction\_num | Person\_id | Pay\_date | Amount |

SUPERVISOR

|  |  |
| --- | --- |
| Person\_id | Dept\_id |

SITE

|  |  |  |
| --- | --- | --- |
| Site\_id | Location | Site\_name |

PRODUCT

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Product\_id | Product\_type | Size | Style | Weight | Product\_price |

SELLS

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Product\_id | Time | Salesman\_id | Customer\_id | Site\_id |

PARTS

|  |  |  |  |
| --- | --- | --- | --- |
| Product\_id | Part\_type | Vendor\_id | Weight |

NUM\_PARTS

|  |  |  |
| --- | --- | --- |
| Product\_id | Part\_type | Num\_parts |

COST\_PARTS

|  |  |  |
| --- | --- | --- |
| Vendor\_id | Part\_type | Part\_cost |

VENDOR

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Vendor\_id | Vendor\_URL | Vendor\_acc\_num | Credit\_rating | Vendor\_addr | Vendor\_name |

**Script to create Physical Schema in Oracle:**

**CREATE TABLE** PERSON

(Fname VARCHAR(30),

Lname VARCHAR(30),

Person\_id NUMBER(15),

Age NUMBER(2),

Sex CHAR(1),

Line\_one VARCHAR(30),

Line\_two VARCHAR(30),

City VARCHAR(30),

State CHAR(2),

Zip NUMBER(5),

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

**CREATE TABLE** PHONE\_NUM

(Person\_id NUMBER(15),

Phone\_num NUMBER(10),

**PRIMARY KEY** (Person\_id, Phone\_num),

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

**CREATE TABLE** CUSTOMER

(Person\_id NUMBER(15),

Pref\_salesman VARCHAR(30),

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

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

**CREATE TABLE** POTENTIAL\_EMPLOYEE

(Person\_id NUMBER(15),

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

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

**CREATE TABLE** DEPARTMENT

(Department\_id NUMBER(15),

Department\_name VARCHAR(30),

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

**CREATE TABLE** CURR\_DEPT

(Dept\_id NUMBER(15),

Person\_id NUMBER(15),

Curr\_dept BOOLEAN,

**PRIMARY KEY** (Department\_id, Person\_id),

**FOREIGN KEY** (Dept\_id) **REFERENCES** DEPARTMENT(Department\_id),

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

**CREATE TABLE** EMPLOYEE

(Person\_id NUMBER(15),

Dept\_id NUMBER(15),

Rank VARCHAR(30),

**PRIMARY KEY** (Person\_id, Dept\_id),

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

**FOREIGN KEY** (Dept\_id) **REFERENCES** DEPARTMENT(Department\_id) );

**CREATE TABLE** APPLICANT

(Person\_id NUMBER(15),

Email VARCHAR(30),

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

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

**CREATE TABLE** SHIFT

(Person\_id NUMBER(15),

Dept\_id NUMBER(15),

Time\_begin TIMESTAMP,

Time\_end TIMESTAMP,

**PRIMARY KEY** (Person\_id, Dept\_id, Time\_begin, Time\_end),

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

**FOREIGN KEY** (Dept\_id) **REFERENCES** DEPARTMENT(Department\_id) );

**CREATE TABLE** JOB

(Job\_id NUMBER(15),

Dept\_id NUMBER(15),

Description VARCHAR(150),

Posted\_date DATE,

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

**FOREIGN KEY** (Dept\_id) **REFERENCES** DEPARTMENT(Department\_id) );

**CREATE TABLE** INTERVIEW

(Job\_id NUMBER(15),

Person\_id NUMBER(15),

Interview\_num NUMBER(1),

Interviewer\_id NUMBER(15),

Interview\_grade CHAR(1),

Interview\_time TIME,

**PRIMARY KEY** (Job\_id, Person\_id, Interview\_num),

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

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

**CREATE TABLE** MONTHLY\_SALARY

(Transaction\_num NUMBER(15),

Person\_id NUMBER(15),

Pay\_date DATE,

Amount DECIMAL(10,2),

**PRIMARY KEY** (Transaction\_num, Person\_id),

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

**CREATE TABLE** SUPERVISOR

(Person\_id NUMBER(15),

Dept\_id NUMBER(5),

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

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

**FOREIGN KEY** (Dept\_id) **REFERENCES** DEPARTMENT(Department\_id) );

**CREATE TABLE** SITE

(Site\_id NUMBER(15),

Location VARCHAR(30),

Site\_name VARCHAR(30),

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

**CREATE TABLE** PRODUCT

(Product\_id NUMBER(15),

Product\_type VARCHAR(30),

Size DECIMAL(10,2),

Style VARCHAR(30),

Weight DECIMAL(10,2),

Product\_price DECIMAL(10,2),

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

**CREATE TABLE** SELLS

(Product\_id NUMBER(15),

Time TIME,

Salesman\_id NUMBER(15),

Customer\_id NUMBER(15),

Site\_id NUMBER(15),

**PRIMARY KEY** (Product\_id, Time, Salesman\_id, Customer\_id, Site\_id),

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

**FOREIGN KEY** (Salesman\_id) **REFERENCES** PERSON(Person\_id),

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

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

**CREATE TABLE** VENDOR

(Vendor\_id NUMBER(15),

Vendor\_URL VARCHAR(30),

Vendor\_acc\_num NUMBER(15),

Credit\_rating NUMBER(15),

Vendor\_addr VARCHAR(30),

Vendor\_name VARCHAR(30),

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

**CREATE TABLE** PARTS

(Product\_id NUMBER(15),

Vendor\_id NUMBER(15),

Part\_type VARCHAR(30),

Weight DECIMAL(10,2),

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

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

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

**CREATE TABLE** PART\_COST

(Vendor\_id NUMBER(15),

Part\_type VARCHAR(30),

Part\_cost DECIMAL(10,2),

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

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

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

**CREATE TABLE** NUM\_PARTS

(Product\_id NUMBER(15),

Part\_type VARCHAR(30),

Num\_parts NUMBER(4),

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

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

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

**VIEWS**

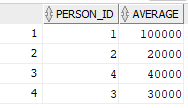
**1**

**CREATE VIEW** VIEW\_ONE **AS**

**SELECT** Person\_id, AVG(amount) as average

**FROM** monthly\_salary

**GROUP BY** person\_id;



**2**

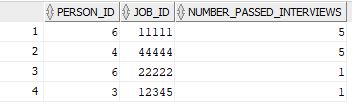
**CREATE VIEW** VIEW\_TWO **AS**

**SELECT** person\_id, job\_id, COUNT (\*) as Number\_passed\_interviews

**FROM** INTERVIEW

**WHERE** interview\_grade > 70

**GROUP BY** person\_id, job\_id;



**3**

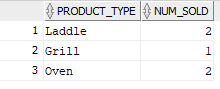
**CREATE VIEW** VIEW\_THREE **AS**

**SELECT** PRODUCT.product\_type, COUNT (\*) as Num\_sold

**FROM** PRODUCT, SELLS

**WHERE** PRODUCT.product\_id = SELLS.product\_id

**GROUP BY** PRODUCT.product\_type;



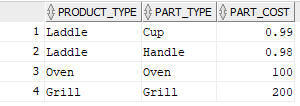
**4**

**CREATE VIEW** VIEW\_FOUR **AS**

**SELECT** PRODUCT.product\_type, PARTS.part\_type, PARTS.part\_cost

**FROM** PARTS, PROD\_PARTS, PRODUCT

**WHERE** PRODUCT.product\_id = PROD\_PARTS.product\_id AND PROD\_PARTS.part\_type = PARTS.part\_type AND PROD\_PARTS.vendor\_id = PARTS.vendor\_id;



**QUERIES**

***1: Return the the ID and Name of interviewers who participate in interviews where the applicant's name is "Hellen Cole" arranged for job "11111"***

SELECT INTERVIEW.interviewer\_id, PERSON.Fname, PERSON.Lname

FROM INTERVIEW, PERSON

WHERE INTERVIEW.job\_id in ( SELECT job\_id

FROM PERSON, INTERVIEW

WHERE PERSON.Fname = 'Hellen' AND PERSON.Lname = 'Cole' AND PERSON.person\_id = INTERVIEW.person\_id AND INTERVIEW.job\_id = 11111) AND interview.interviewer\_id = person.person\_id;

***Output:***

INTERVIEWER\_ID FNAME LNAME

-------------- ------------------------------ ------------------------------

2 Camryn Rogers

3 Rob Rogers

2 Camryn Rogers

3 Rob Rogers

2 Camryn Rogers

2.

***# 2: Return the ID of all jobs which are posted by department "Marketing" in January, 2011***

SELECT JOB.job\_id

FROM JOB, DEPARTMENT

WHERE JOB.dept\_id = department.department\_id AND department.department\_name = 'Marketing' AND EXTRACT(MONTH FROM job.posted\_date) = 1 AND EXTRACT(YEAR FROM job.posted\_date) = 2011;

***Output:***

JOB\_ID

----------

11111

22222

33333

***# 3: Return the ID and name of the salesmen with no supervisees***

SELECT person.person\_id, person.fname, person.lname

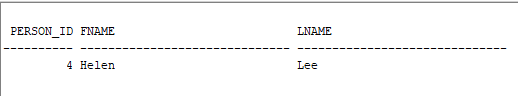
FROM person, supervisor

WHERE person.person\_id = supervisor.person\_id AND supervisor.dept\_id NOT IN (SELECT supervisor.dept\_id

FROM CURR\_DEPT, SUPERVISOR

WHERE curr\_dept.dept\_id = supervisor.dept\_id AND CURR\_DEPT.person\_id != supervisor.person\_id AND curr\_dept.curr\_dept = 'Y');

***Output:***



***# 4 : Return the ID and Location of the Marketing sites which have no sale records during March 2011***

SELECT DISTINCT SITE.site\_id, SITE.location

FROM JOB, DEPARTMENT, SITE

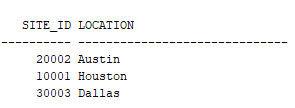
WHERE SITE.site\_name = 'Marketing' AND NOT EXISTS(

SELECT \*

FROM JOB, DEPARTMENT

WHERE job.dept\_id = department.department\_id AND department.department\_name = 'Marketing' AND EXTRACT(MONTH FROM job.posted\_date) = 3 AND EXTRACT(YEAR FROM job.posted\_date) = 2011);

***Output:***



***# 5 : Return the job's id and description which does not hire a suitable person one month after it is posted***

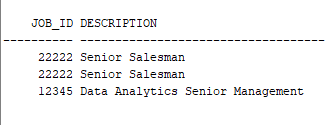
SELECT job.job\_id, job.description

FROM job JOIN interview ON job.job\_id = interview.job\_id

GROUP BY interview.person\_id, job.job\_id, job.description

HAVING COUNT(\*) != 5 OR AVG(interview\_grade) < 70;

***Output:***



***# 6 : Return the ID and name of the salesmen who have sold all products that cost over $200***

SELECT DISTINCT product.product\_id

FROM product JOIN sells ON product.product\_id = sells.product\_id

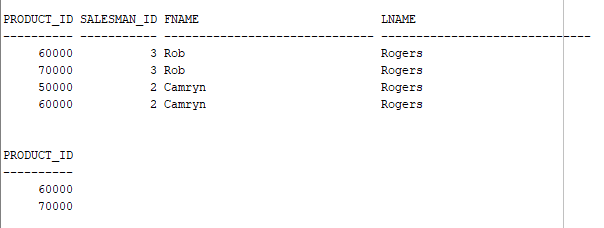
WHERE product\_price > 200.00;

SELECT sells.product\_id, sells.salesman\_id, person.fname, person.lname

FROM sells, person

WHERE person.person\_id = sells.salesman\_id

GROUP BY salesman\_id, product\_id, person.fname, person.lname;

***Output:***

\*I couldn’t figure out how to get the answer in one statement, so the top one is the product that each salesman sold and the bottom one lists the product ID’s that are over $200.00. So the answer to the query would be:

**3 Rob Rogers**

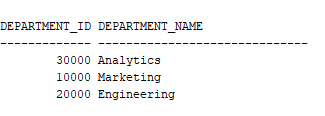
***# 7 : Return the department ID and name which has no job posting during 1/1/2011 and 2/1/2011***

SELECT DISTINCT department.department\_id, department.department\_name

FROM JOB, DEPARTMENT

WHERE JOB.dept\_id = department.department\_id AND job.posted\_date != TO\_DATE('2011-01-01','YYYY-MM-DD') AND job.posted\_date != TO\_DATE('2011-02-01','YYYY-MM-DD');

***Output:***



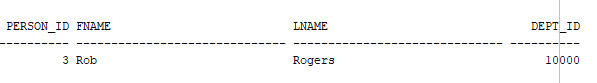
***# 8 : Return the ID, name, and dept ID of existing employees who apply to job '12345'***

SELECT person.person\_id, person.fname, person.lname, curr\_dept.dept\_id

FROM person, curr\_dept, employee, applicant, interview

WHERE person.person\_id = employee.person\_id AND employee.person\_id = applicant.person\_id AND person.person\_id = curr\_dept.person\_id AND curr\_dept = 'Y' AND person.person\_id = interview.person\_id AND interview.job\_id = 12345;

***Output:***



***# 9 : Return the best seller's type in the company (sold the most items)***

SELECT product\_type

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

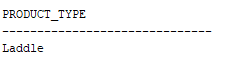
FROM product

GROUP BY product\_type

ORDER BY COUNT(product\_type) DESC)

WHERE ROWNUM = 1

***Output:***



***# 10: Return the product type whose net profit is highest***

SELECT product\_type

FROM (SELECT product.product\_type, product.product\_price, sum(num\_parts \* part\_cost) , (product.product\_price - sum(num\_parts \* part\_cost))

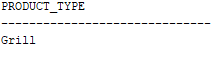
FROM product JOIN prod\_parts ON product.product\_id=prod\_parts.product\_id JOIN parts ON prod\_parts.vendor\_id=parts.vendor\_id AND prod\_parts.part\_type=parts.part\_type

GROUP BY product.product\_type,product.product\_price, product.product\_id

ORDER BY product.product\_price - sum(num\_parts \* part\_cost) DESC)

WHERE ROWNUM = 1;

***Output:***



***# 11: Return the name and id of the employees who has worked in all departments***

SELECT person\_id, fname, lname

FROM person

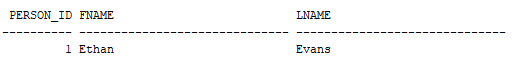
WHERE person\_id IN (SELECT curr\_dept.person\_id

FROM curr\_dept JOIN department ON curr\_dept.dept\_id = department.department\_id

GROUP BY curr\_dept.person\_id

HAVING COUNT(\*) = (SELECT COUNT(\*)FROM department));

***Output:***



***# 12: Return the name and email addresses of the interviewee who is selected***

SELECT person.fname, person.lname, applicant.email

FROM person, applicant

WHERE person.person\_id IN (

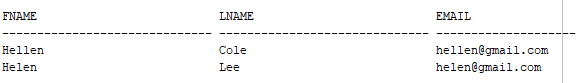
SELECT interview.person\_id

FROM interview

GROUP BY interview.person\_id,interview.job\_id

HAVING COUNT(\*) = 5 AND AVG(interview.interview\_grade) > 70.00) AND person.person\_id = applicant.person\_id;

***Output:***



\*In the instructions it says “employee who is accepted” – Hellen Cole and Helen Lee were both selected for different job\_id’s (11111 and 44444)

***# 13: Retreive the name, phone number, email address of the interviewees selected for all the jobs that they applied for***

SELECT person.fname, person.lname, phone\_num.phone\_num, applicant.email

FROM person, applicant, phone\_num

WHERE person.person\_id IN (

SELECT interview.person\_id

FROM interview

GROUP BY interview.person\_id,interview.job\_id

HAVING COUNT(\*) = 5 AND AVG(interview.interview\_grade) > 70.00) AND person.person\_id NOT IN(SELECT interview.person\_id

FROM interview

GROUP BY interview.person\_id,interview.job\_id

HAVING COUNT(\*) != 5 OR AVG(interview.interview\_grade) < 70.00) AND person.person\_id = applicant.person\_id AND phone\_num.person\_id = applicant.person\_id;

***Output:***



***# 14: Return the employee's name and id whose average monthly salary is highest in the company***

SELECT person\_id, fname, lname

FROM person

WHERE person\_id = (

SELECT person\_id

FROM (SELECT Person\_id, AVG(amount)

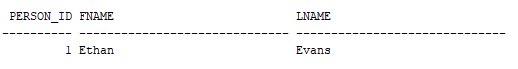
FROM monthly\_salary

GROUP BY person\_id

ORDER BY AVG(amount DESC)

WHERE ROWNUM = 1);

***Output:***



***# 15: Return the ID and name of the vendor who supplies part 'Cup' and weight is less than 4 lbs and price is lowest amoung all vendors***

SELECT vendor\_id, vendor\_name

FROM vendor

WHERE vendor\_id = (

SELECT vendor\_id

FROM (SELECT vendor\_id, MIN(part\_cost)

FROM parts

WHERE part\_type = 'Cup' AND weight < 4.0

GROUP BY vendor\_id

ORDER BY MIN(part\_cost) ASC)

WHERE ROWNUM = 1);

***Output:***

