Pharmaceutical Representative Database

Overview: This is a database of pharmaceutical company, a prominent manufacturer of drugs and vaccines for immunology, oncology, cardiology, dialectology/endocrinology, and neurology. This database stores record of medical sales representatives, drugs and vaccine samples and their distribution.

The company has numbers of sales representatives for the distribution of their products. A medical representative will have number of sample drugs and medical products from the company. Those samples are distributed to doctors, clinics, hospitals, pharmacy etc. There will be a dashboard for each sales rep where they could track the progress of the work allocated and the number of samples distributed, expiry dates of the samples and so on. Also, the manager can monitor the performance of sales reps and depending on the performance each sales rep can be rewarded or will be given appraisal.

Assumptions:

1. The system stores data about medical sales representatives, doctors, clinics, product(drugs and vaccines), pharmacy
2. The sales representatives are categorized according to the specialization.
3. Sales representatives will be assigned on different locations and will have meetings with doctors and clinics on those locations. Each day a sales representative will have meetings with doctors or clinics.
4. Each sales representative must distribute 500 samples per product each month and if the sales exceeds the limit, Sales representative can place an order for more samples with the consent of higher authority.
5. Main worksheet will store the target sales of the year, dates for meetings and follow up meetings with potential clients. Based on the main worksheet, each sales representative will be assigned weekly timesheet.
6. The company can hire and fire sales representatives. Sales representatives may be part-time or full-time. Each sales representative will have benefits as phone bill coverage, car payment, car insurance, fuel payment.
7. Each sales representative has a name, gender, birthdate, SSN, address, phone, email, salary, benefits, qualification, work experience, hiredate, date of leaving, status.
8. Each doctor and clinic has a name, gender(for individual), address, phone, email, speciality, License status, history of previous purchases if any.
9. The database will record the number of sales per sales representative, number of new customers per sales representative, level of performance during conferences and how successful was conference and meetings. And based on these factors each sales

representative will be evaluated and will receive bonuses, salary raise or promotion at the end of the year.

10. Doctors will review and rate each product, report side effects and efficacy. Products with good reviews and ratings will be nominated for manufacturing and products with rating below ⅖ will be temporarily shut down and transferred to the failure list, with the consent of the higher authority the samples might be sent back to lab for re-evaluation where the company may manufacture a revised version of the product.
11. Products will have a name, manufacture date, expiry date, ingredients, warnings, dosage.

Features:

Entering data:

1. Allow a user(sales representative) or an administrator(manager) to register and create his or her username and password. The user can also modify the password by entering username, old password, and new password. The user can also modify other personal information.
2. Validate the identity of a user or administrator based on username and password.
3. Allow the manager to create a worksheet and assign to each sales representative.
4. Allow the manager to insert detailed record of the new medicine samples. Also modify or delete the records of the medicine samples.
5. Sales representative can update the tasks if completed or can modify the dates of any particular task with the consent of the manager.
6. Sales representative needs to update the information like name of the sample, number of samples, date of distribution etc. distributed to each doctor.
7. Allow the administrator to insert detailed record of the new sales representative.
8. Allow the administrator to modify record of the sales representative.

Compute statistics:

9. Compute ranking of Sales representatives based on the highest sales, no. of new clients persuaded by sales representative, no. of conferences, “lunch and learn”, “breakfast and learn” conducted or attended and their level of performance in those meetings **for each year**. This ranking will be used for employee evaluation purpose. Top 5 Sales representatives will receive extra benefits and recognition and for bottom 5, their department(specialization) manager will either assign them advance training or will fire them.
10. Compute ranking of Sales representatives based on the highest sales, no. of new clients persuaded by sales representative, no. of conferences, “lunch and learn”, “breakfast and learn” conducted or attended and their level of performance in those meetings **for entire career** before promoting them.
11. Compute ranking of each department(specialization) based on the highest sales, no. of new clients persuaded by sales representative.

12. Compute the budget estimation for catering and conducting conferences.
13. Compute the performance of a sales representative for each quarter. Each sales representative are supposed to do better than previous quarter.
14. Return the number of sales by product for each location covered by a sales representative.
15. Compute the date 3 days prior to the date of expiry of product to notify doctors and sales representative.
16. Compute availability of the day in work schedule while assigning the task to a sales representative or while changing the dates of the particular tasks.
17. Compute system's ratings for medicine samples by computing the count of doctors to whom the samples were distributed, client's rating given to the samples. If system rating of the sample is more than 4 then medicine sample gets qualified for manufacturing list.

ERD Assumption:

- Each department will have one manager and the manager will manage sales representatives of that department. (Only names of manager has been recorded as other information of manager are of no importance to our project)

- Sales representative and doctors will have one user account.

- Sales representatives might will have appointments with doctors for sample distributions.

- With the agreement of doctor, various product samples might be distributed by sales rep on the same meeting and quantity for each distribution will be recorded separately.

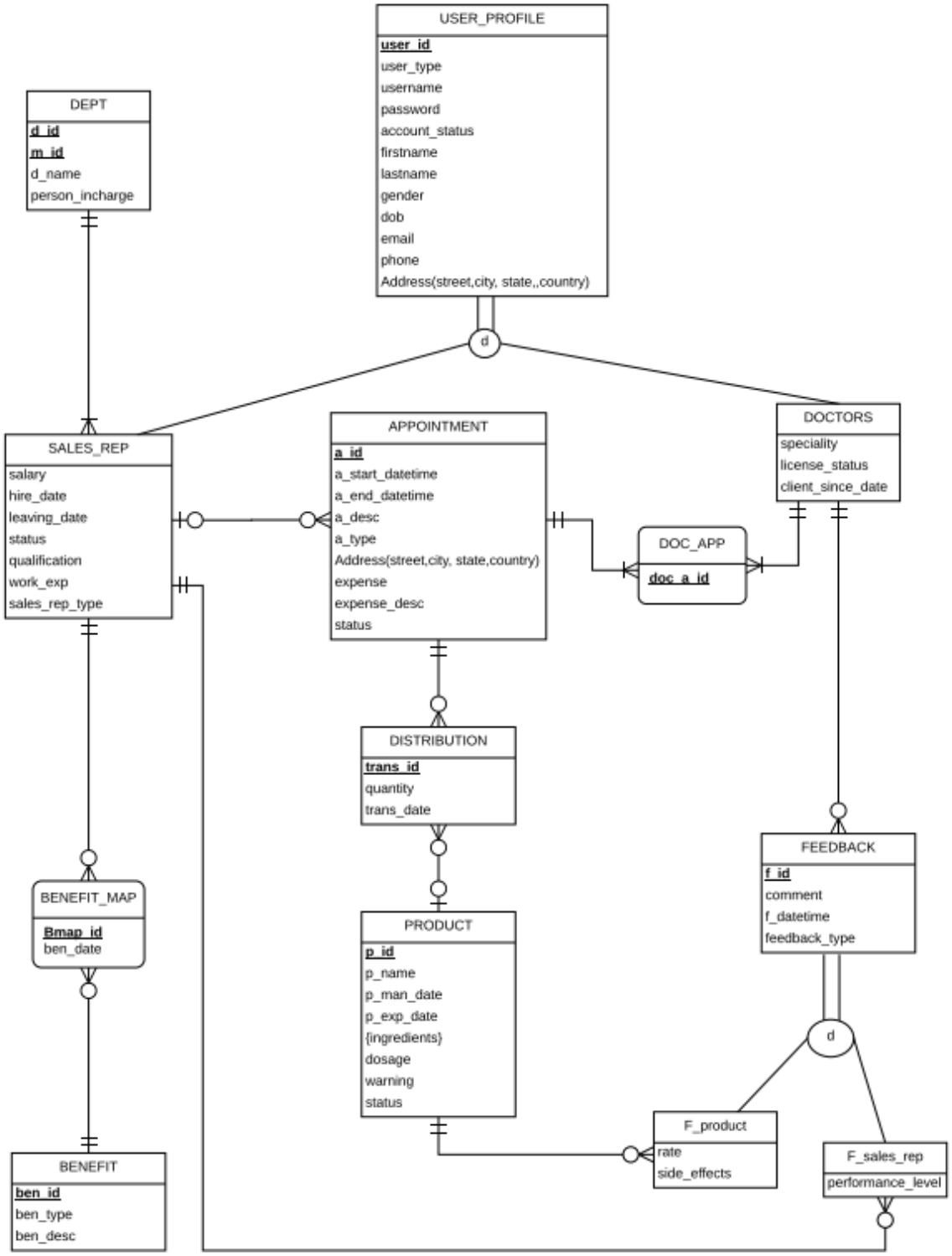
- Appointment table will record meetings and conferences as well. In one meeting or conference there can be more than one doctor, and each doctor present in the certain appointment will be recorded in doc_app table.

- After appointment, doctors can give feedback about sales reps and products. Performance level for sales rep and rate, side effects for product samples might be given by doctors.

- s_id in doctors table records the ID of sales rep who brought in the doctor as client of the company.

- Various benefits might will be given to sales reps (based on their performance). Part time sales rep will not receive any kind of benefits and will have almost have of avg salary of full time employees(there will be no other difference than full_time Sales rep)

ERD:


USER_PROFILE
user_id
user_type
username
password
account_status
firstname
lastname
gender
dob
email
phone
Address(street,city, state,,country)
d
DEPT
d_id
m_id
d_name
person_incharge
SALES_REP
salary
hire_date
leaving_date
status
qualification
work_exp
sales_rep_type
APPOINTMENT
a_id
a_start_datetime
a_end_datetime
a_desc
a_type
Address(street,city, state,country)
expense
expense_desc
status
DOCTORS
speciality
license_status
client_since_date
DOC_APP
doc_a_id
DISTRIBUTION
trans_id
quantity
trans_date
PRODUCT
p_id
p_name
p_man_date
p_exp_date
{ingredients}
dosage
warning
status
FEEDBACK
f_id
comment
f_datetime
feedback_type
d
F_product
rate
side_effects
F_sales_rep
performance_level
BENEFIT_MAP
Bmap_id
ben_date
BENEFIT
ben_id
ben_type
ben_desc

Create Statements:

PRODUCT

```
create table product(
p_id int PRIMARY KEY NOT NULL,
p_name varchar(255),
p_man_date date,
p_exp_date date,
ingredients varchar(255),
dosage varchar(255),
warning varchar(255),
status varchar(20));
```

DEPARTMENT

```
create table department(
d_id int,
m_id int,
d_name varchar(30),
person_incharge varchar(50),
PRIMARY KEY(d_id, m_id));
```

USER PROFILE

```
create table user_profile(
user_id int PRIMARY KEY NOT NULL,
username varchar(30),
password varchar(30),
user_type char(1) CHECK(user_type IN('S', 'D')),
account_status char(1) CHECK(account_status IN(0,1)) ,
firstname varchar(30),
lastname varchar(30),
gender varchar(10),
dob date,
addr_street varchar(55),
addr_city varchar(30),
addr_state char(2),
addr_country varchar(30),
phone int,
email varchar(50),
CONSTRAINT unique_username UNIQUE(username),
CONSTRAINT unique_usertype UNIQUE(user_id, user_type));
```

SALES REPS

```
create table sales_rep(
s_id int PRIMARY KEY NOT NULL,
user_type char(1) DEFAULT 'S' NOT NULL,
CHECK(user_type = 'S'),
s_salary decimal,
hire_date date,
leaving_date date,
status char(1) CHECK(status in(0,1)),
qualification varchar(255),
work_exp int,
s_type char(9) CHECK(s_type IN ('full_time', 'part_time')),
FOREIGN KEY(s_id) REFERENCES user_profile(user_id),
d_id int,
m_id int,
FOREIGN KEY(d_id, m_id) REFERENCES department(d_id,m_id),
UNIQUE(s_id,user_type));
```

DOCTORS

```
create table doctors(
doc_id int PRIMARY KEY NOT NULL,
user_type char(1) DEFAULT 'D' NOT NULL,
CHECK(user_type = 'D'),
speciality varchar(255),
license_status char(1) CHECK(license_status in(0,1)),
client_since_date date,
s_id int,
FOREIGN KEY(s_id) REFERENCES sales_rep(s_id),
FOREIGN KEY(doc_id) REFERENCES user_profile(user_id),
UNIQUE(doc_id, user_type));
```

APPOINTMENTS

```
create table appointment(
a_id int PRIMARY KEY NOT NULL,
a_start_time timestamp,
a_end_time timestamp,
a_addr_street varchar(55),
a_addr_city varchar(30),
a_addr_state char(2),
a_addr_country varchar(30),
a_desc varchar(255),
s_id,
FOREIGN KEY(s_id) REFERENCES sales_rep(s_id),
expense int,
expense_desc varchar(255),
status char(1) CHECK(status IN(1,0)));
```

DOCTOR APPOINTMENT

```
create table doc_appointment(
a_id int,
doc_id int,
PRIMARY KEY(a_id, doc_id),
FOREIGN KEY(a_id) REFERENCES appointment(a_id),
FOREIGN KEY(doc_id) REFERENCES doctors(doc_id));
```

DISTRIBUTION

```
create table distribution(
trans_id int PRIMARY KEY NOT NULL,
trans_date date,
quantity int,
p_id int,
FOREIGN KEY(p_id) REFERENCES product(p_id),
a_id int,
FOREIGN KEY(a_id) REFERENCES appointment(a_id));
```

BENEFIT

```
create table benefit(
ben_id int PRIMARY KEY NOT NULL,
ben_type varchar(50),
ben_desc varchar(255));
```

BENEFIT MAP

```
create table benefit_map(
s_id int,
ben_date date,
ben_id int,
FOREIGN KEY(s_id) REFERENCES sales_rep(s_id),
FOREIGN KEY(ben_id) REFERENCES benefit(ben_id));
```

FEEDBACK

```
create table feedback(
f_id int PRIMARY KEY NOT NULL,
f_comment varchar(255),
f_datetime timestamp,
f_type char(1) NOT NULL CHECK(f_type IN('P', 'S')),
UNIQUE(f_id, f_type),
doc_id int,
FOREIGN KEY(doc_id) REFERENCES doctors(doc_id));
```

FEEDBACK SALES REPS

```
create table fb_sales_rep(
f_id int PRIMARY KEY NOT NULL,
f_type char(1) DEFAULT 'S' NOT NULL,
```

```
CHECK(f_type = 'S'),
performance_level varchar(20) CHECK(performance_level IN ('great', 'good', 'average', 'bad')),
FOREIGN KEY(f_id) REFERENCES feedback(f_id),
FOREIGN KEY(f_id) REFERENCES feedback(f_id),
UNIQUE(f_id, f_type),
s_id,
FOREIGN KEY(s_id) REFERENCES sales_rep(s_id)
);
```

FEEDBACK PRODUCT

```
create table fb_product(
f_id int PRIMARY KEY NOT NULL,
f_type char(1) DEFAULT 'P' NOT NULL,
CHECK(f_type = 'P'),
rate int CHECK (rate IN (1, 2, 3, 4, 5)),
side_effect varchar(255),
FOREIGN KEY(f_id) REFERENCES feedback(f_id),
FOREIGN KEY(f_id) REFERENCES feedback(f_id),
p_id int,
FOREIGN KEY(p_id) REFERENCES product(p_id),
UNIQUE(f_id, f_type)
);
```

Insert Queries:

DEPARTMENT

```
insert all
into department values(101,1, 'Allergy and Immunology','Sanil Patankar')
into department values(102,2, 'Dermatology','Vasu Misal')
into department values(103,3, 'Pediatrics','Suraksha Shukla')
into department values(104,4, 'Family Practice','Sanil_S Patankar')
into department values(105,5, 'Otorhinolaryngology','Vasu_N Misal')
into department values(106,6, 'Geriatric Medicine','Suraksha_A Shukla')
select * from dual;
```

USER PROFILE

```
insert all
into user_profile values(1,'JamesBlunt','JamesBlunt','S',1,'James','Blunt','Male',date'1986-03-04','1234 Street','Baltimore','MD','USA',4436544567,'JamesBlunt@gmail.com')
into user_profile values(2,'JeffBlue','JeffBlue','S',1,'Jeff','Blue','Male',date'1989-06-08','1276 Street','Columbia','MD','USA',4436456567,'JeffBlue@gmail.com')
into user_profile values(3,'JackBlack','JackBlack','S',1,'James','Black','Male',date'1984-12-9','12776 Street','Baltimore','MD','USA',4436576987,'JackBlack@gmail.com')
into user_profile values(4,'AmyHack','AmyHack','S',1,'Amy','Hack','Female',date'1986-4-3','127 Street','Ellicot','MD','USA',4436545467,'AmyHack@gmail.com')
into user_profile values(5,'EmaBlaze','EmaBlaze','S',1,'Ema','Blaze','Female',date'1989-7-7','1233 Street','Baltimore','MD','USA',4556856567,'EmaBlaze@gmail.com')
into user_profile values(6,'JesseJay','JesseJay','S',1,'Jesse','Jay','Female',date'1989-12-14','1212 Street','Towson','MD','USA',4490066987,'JesseJay@gmail.com')
into user_profile values(7,'RobertDowny','RobertDowny','D',1,'Robert','Downy',
'Male',date'1979-3-7','123A Belkin','Baltimore','MD','USA',4416676789, 'RobertD@gmail.com')

into user_profile values(8 ,'RobertPattinson','RobertPattinson','D', 1,'Robert','Pattinson','Male',
date'1969-03-17','12342 Ave','Ellicot','MD','USA',3440986789,'RobertPattinson@gmail.com')
into user_profile values(9,'AceHarper','AceHarper','D',1,'Ace','Harper','Male',date'1972-06-27','123E Old','Towson','MD','USA',4410986569,'AceHarper@gmail.com')
into user_profile values(10,'AlisonGrey','AlisonGrey','D',1,'Alison','Grey','Female',date'1974-01-07','123WE West','Columbia','MD','USA',4414586789,'AlisonGrey@gmail.com')
into user_profile values(11,'MayHalley','MayHalley','D',1,'May','Halley','Female',date'1974-07-23','1231G Newdale','Baltimore','MD','USA',4789986789,'MayHalley@gmail.com')
into user_profile values(12,'KellyBrown','KellyBrown','D',1,'Kelly','Brown','Female',date'1979-05-13','123 Street','Halethorpe','MD','USA',4410086789,'KellyBrown@gmail.com')
select * from dual;
```

PRODUCTS

```
insert all
into product values(1,'Carbinoxamine',date '2015-6-3',date '2017-6-3','ingredients','1 or 2 tablets (4 to 8 mg) 3 to 4 times daily','Caution should be exercised in patients with history of asthma, glaucoma, difficulty in urinating, stomach or bladder obstruction, prostate problems',1)
into product values(2,'Cetrizine',date '2015-7-4',date '2016-7-4','ingredients','The recommended dose is 10 mg once daily','Caution should be exercised in patients with history of kidney or liver',1)
into product values(3,'Dimethindene',date '2015-4-9',date '2017-4-9','ingredients','1-2 mg 3 times-day. Topical As 0.1% gel-lotion: Use as directed','Caution should be exercised in patients with history of elderly, increased eye pressure, prostate enlargement, epilepsy',1)
into product values(4,'Ebastine',date '2015-11-14',date '2018-11-14','ingredients','The recommended dose is 10-20 mg once daily','Caution should be exercised in patients with history of liver and kidney impairment, QT interval prolongation, during pregnancy and breastfeeding',1)
into product values(5,'Ketotifin',date'2015-8-24',date'2017-8-24','ingredients','1 mg twice daily,
```

up to 2 mg twice daily if needed','It may cause dizziness or drowsiness, do not drive a car or operate machinery while taking this medication',1)
into product values(6,'Methdilazine',date'2015-3-15',date'2016-3-15','ingredients','8-16 mg twice daily','It may cause drowsiness or blurred vision, do not drive a car or operate machinery while taking this medication',1)
select * from dual;

SALES_REPS

insert all
into sales_rep values(1,'S',45000,date '2014-6-5',NULL,1,'Certification in sales',24,'full_time',101,1)
into sales_rep values(2,'S',40000,date '2014-5-2',NULL,1,'Certification in sales',20,'full_time',102,2)
into sales_rep values(3,'S',48000,date '2013-11-25',NULL,1,'Diploma in sales',29,'full_time',103,3)
into sales_rep values(4,'S',45000,date '2014-5-9',NULL,1,'Certification in sales',24,'full_time',104,4)
into sales_rep values(5,'S',40000,date '2014-5-2',NULL,1,'Certification in sales',20,'full_time',105,5)
into sales_rep values(6,'S',40000,date '2014-11-25',NULL,1,'Certification in sales',20,'full_time',106,6)
select * from dual;

DOCTORS

insert all
into doctors values(7,'D','Allergy and Immunology',1,date '2013-3-2',1)
into doctors values(8,'D','Dermatology',1,date'2013-4-3',2)
into doctors values(9,'D','Pediatrics',1,date'2013-5-6',3)
into doctors values(10,'D','Family Practice',1,date'2013-7-7',4)
into doctors values(11,'D','Otorhinolaryngology',1,date'2013-9-9',5)
into doctors values(12,'D','Geriatric Medicine',1,date'2013-4-4',6)
select * from dual;

APPOINTMENT

insert all
into appointment values(1, timestamp '2015-11-11 13:30:00', timestamp '2015-11-11 15:30:00', '8556 Street','Columbia','MD','USA','Promotion of Carbinoxamine', 1, 4000, 'travel=$2000, hotel and food=$2000', 1)
into appointment values(2, timestamp '2015-11-12 10:00:00', timestamp '2015-11-12 11:30:00','177 Street','Elkridge','MD','USA', 'Promotion of Cetrizine', 2, 5000, 'travel=$3000, hotel and food=$2000', 1)

```
into appointment values(3, timestamp '2015-11-12 10:30:00',  timestamp '2015-11-12 12:00:00','807 Street','Laurel','MD','USA', 'Promotion of Dimethindene', 3,  4500, 'travel=$2000, hotel and food=$2500', 1)
into appointment values(4, timestamp '2015-11-11 13:30:00',  timestamp '2015-11-11 14:30:00','773 Street','Catonsville','MD','USA', 'Promotion of Advil', 4, 3000, 'travel=$1000, hotel and food=$2000', 1)
into appointment values(5, timestamp '2015-11-12 10:00:00',  timestamp '2015-11-12 11:30:00','226 Street','Baltimore','MD','USA', 'Promotion of Brufen', 5, 3500, 'travel=$2000, hotel and food=$1500', 0)
into appointment values(6, timestamp '2015-11-12 10:30:00',  timestamp '2015-11-12 12:00:00', '807 Street','Frederick','MD','USA','Promotion of Tylenol', 6, 3800, 'travel=$2000, hotel and food=$1800', 0)
select * from dual;
```

DOC_APPOINTMENT

```
insert all
into doc_appointment values(1, 7)
into doc_appointment values(2, 8)
into doc_appointment values(3, 9)
into doc_appointment values(4, 10)
into doc_appointment values(5, 11)
into doc_appointment values(6, 12)

select * from dual;
```

BENEFIT

```
insert all
into benefit values(1,'Fringe Benefits','These may include tuition assistance, flexible medical or child-care spending accounts')
into benefit values(2,'Paid Time Off','Paid time off are holidays, sick leave, and vacation leave.')
into benefit values(3,'Disability Insurance','Covers all income that is lost when a worker is unable to perform their job because of illness or injury')
into benefit values(4,'Medical Insurance','Medical insurance covers the costs of physician and surgeon fees, hospital rooms, and prescription drugs')
into benefit values(5,'Phone bill coverage','All phone bills will be covered')
into benefit values(6,'Gas bill coverage','All gas bills will be covered')
select * from dual;
```

BENEFIT MAP

```
insert all
into benefit_map values(1,date '2015-10-04',1)
into benefit_map values(2,date '2015-09-24',2)
into benefit_map values(3,date '2015-8-14',3)
into benefit_map values(4,date'2015-10-5',4)
```

```
into benefit_map values(5,date'2015-9-5',1)
select * from dual;
```

DISTRIBUTION

```
insert all
into distribution values(1,date'2015-4-10',20,1,1)
into distribution values(2,date'2015-4-15',10,2,2)
into distribution values(3,date'2015-4-11',10,3,3)
into distribution values(4,date'2015-5-19',15,4,4)
into distribution values(5,date'2015-5-1',10,5,5)
into distribution values(6,date'2015-9-12',20,6,6)

select * from dual;
```

FEEDBACK

```
insert all
into feedback values(1, 'product looks promising', timestamp '2015-09-11 14:30:00', 'P', 7)
into feedback values(2, 'definitely would recommend your product', timestamp '2015-09-11
14:30:00', 'S', 7)
into feedback values(3, 'good product, keep up', timestamp '2015-09-12 11:30:00', 'P', 8)
into feedback values(4, 'definitely would recommend your product', timestamp '2015-09-12
11:30:00', 'S',8)
into feedback values(5, 'product looks promising', timestamp '2015-09-12 12:00:00', 'P', 9)
into feedback values(6, 'product looks promising', timestamp '2015-09-12 12:00:00', 'S', 9)
into feedback values(7, 'product looks promising', timestamp '2015-09-11 14:30:00', 'P', 10)
into feedback values(8, 'definitely would recommend your product', timestamp '2015-09-11
14:30:00', 'S', 10)
into feedback values(9, 'good product, keep up', timestamp '2015-09-12 11:30:00', 'P', 11)
into feedback values(10, 'definitely would recommend your product', timestamp '2015-09-12
11:30:00', 'S',11)
into feedback values(11, 'product looks promising', timestamp '2015-09-12 12:00:00', 'P', 12)
into feedback values(12, 'product looks promising', timestamp '2015-09-12 12:00:00', 'S', 12)
select * from dual;
```

FB_PRODUCT

```
insert all
into fb_product values(1, 'P', 4, 'Dizziness or severe sleepiness', 1)
into fb_product values(3, 'P', 3, 'swelling in your face or hands', 2)
into fb_product values(5, 'P', 5, 'None', 3)
into fb_product values(7, 'P', 4, 'Dizziness or severe sleepiness', 4)
```

```
into fb_product values(9, 'P', 3, 'swelling in your face or hands', 5)
into fb_product values(11, 'P', 5, 'None', 6)
select * from dual;
```

FB_SALES_REPS

```
insert all
into fb_sales_rep values(2, 'S', 'good', 1)
into fb_sales_rep values(4, 'S', 'average', 2)
into fb_sales_rep values(6, 'S', 'great', 3)
into fb_sales_rep values(8, 'S', 'good', 4)
into fb_sales_rep values(10, 'S', 'average', 5)
into fb_sales_rep values(12, 'S', 'great', 6)
select * from dual;
```

Feature Specification:

The manager id(m_id) indicates the manager of the employee and department id(d_id) refers to the department which the employee belongs.

With the use of primary key(s_id) sales rep information can be use for setting up an appointments with the doctor or keeping a track of their performance.
The output will be a table containing all the information mentioned.

1. The input for registration of sales rep will include personal information(firstname, lastname, gender, dob, addr_street, addr_city, addr_state, addr_country, phone, email) into **user_profile** table.
2. The registration of sales rep also includes the login credentials i.e username(username), password(password) and user id(user_id) which will authenticate the user. Depending on the type of user various functionalities shall be available. The username and userID & usertype are unique constraints.
3. The manager will assign tasks to the employees of his department. In order to complete the task, the sales rep can take appointments with the doctors and set up meetings, where the sales rep will talk about the new medicine and distribute samples.
4. Managers will insert new records of product(p_id, p_name, p_man_date, p_exp_date, ingredients, dosage, warning, status)
   Also manager will modify some data of product such as (dosage, warning, status).
   They can also delete records (Delete From product where pname = ‘user given value’;)
5. Once sales representative has completed task then s/he will update appointment status, appointment end date time, expense, expense description and number of samples distributed(a_status = ‘1’ for completed appointment, a_end_datetime, expense, expense_desc, quantity)
6. ID of the product, quantity of that product and date of distribution will be inserted for each product distributed to doctors by the sales representative.

7. Other details of sales representative as salary, hire_date, work_experience, qualification, department ID will be inserted in **sales_rep** table. Leaving_date will be NULL and status will be 1(1 being current sales rep of the company) during insertion.
8. In case if sales rep leaves or gets salary raise, then columns like salary, leaving_date and status will be updated. [status will be changed to 0, 0 - no more employee of the company]
9.
    a. Sales representatives will be ranked based up on the quantity of sample distribution, for which 'quantity' of distribution will be computed from 'distribution' table.
    b. Sales representatives will also be ranked based on the no. of new clients, for which count of 'client_since_date' in 'doctors' table will give number of new doctors and 's_id' will give ID of sales representative who brought in the doctor.
    c. Sales representatives will also be ranked based up on the feedback given by doctors after appointment which will be calculated using 'performance_level' from fb_sales_rep table.
10. These calculations will be done for each year performance and also since the sales representative was hired. For this time interval and hiredate of sales representative will be used.
11. Number of sales by sales representatives of each department will also be calculated for which d_id(department id) in 'sales_rep' table and 'quantity' of distribution of sample by that sales reps in distribution table will be used.
12. Budget will be estimated from appointment table using average of expense column for a particular city.
13. Performance level of each sales rep will be computed based on no. of sales, feedback of performance level, no of new clients as mentioned above in (9) for each quarter and each sale reps are expected to do better than previous quarter.
14. Quantity of distribution for each product will be computed for each location using **quantity** column from **distribution** table and location(**city** or **state**) from **appointment** table
15. Doctors and/or sales representatives will be notified of expiry date before three days for a product by using **expiry_date** from **product** table and using interval.
16. Availability of a sales rep will be computed by using **start_datetime** and **end_datetime** of appointment for that day from **appointment** table, if sales rep doesn't have any appointment at that time then only new task can be assigned. [new task will be inserted in appointment table with status 0]

17. Rating of samples will be computed using average of all ratings given by doctors using **rate** column from **fb_product** table.

Deliverable 4

Code:

1, 2 & 7. The input for registration of sales rep will include personal information(firstname, lastname, gender, dob, addr_street, addr_city, addr_state, addr_country, phone, email) into **user_profile** table.

The registration of sales rep also includes the login credentials i.e username(username), password(password) and user id(user_id) which will authenticate the user. Depending on the type of user various functionalities shall be available. The username and userID & usertype are unique constraints.

Other details of sales representative as salary, hire_date, work_experience, qualification, department ID will be inserted in **sales_rep** table. Leaving_date will be NULL and status will be 1(1 being current sales rep of the company) during insertion.

Solution:-

```
CREATE or REPLACE PROCEDURE insert_user(u_type in char, username_ip in varchar2,
pswd in varchar2, fname in VARCHAR2, lname in varchar2, gen in char, birthdate in date,
add_s in VARCHAR2, add_c in varchar, add_state in char, add_country in VARCHAR2, phn in
number, mailid in VARCHAR2, sal in number, hire in date, leaving in date, qual in
VARCHAR2, exp in number, type_s in char, dep in number, manag in number,
spec in VARCHAR2, repid in number)
is
max_uid number; /*1. to store user_id, since it's auto-incremented and references to s_id and
d_id in sales rep and doctor table*/
begin
SELECT count(*) into max_uid from USER_PROFILE;
INSERT into USER_PROFILE values(max_uid +1, username_ip, pswd, u_type,'1',fname,
lname, gen, birthdate, add_s, add_c, add_state, add_country, phn, mailid);
if (u_type = 'S') then --if user is sales representative
```

```
INSERT into SALES_REP  values(max_uid + 1 , u_type, sal, hire, leaving,'1', qual,exp, type_s,
dep,manag); --refers to comment 1.
dbms_output.put_line('Information Added for Sales Representative: '||(max_uid+1));
elsif (u_type = 'D') then --if user is doctor
INSERT into DOCTORS values(max_uid + 1, u_type,spec,'1',sysdate,repid); --refers to
comment 1.
dbms_output.put_line('Information Added for Doctors: '||(max_uid+1));
else
 dbms_output.put_line('User type error');
end if;
dbms_output.put_line('Successful');
end;
```

Before executing procedure:

```
select * from user_profile;
select * from sales_rep;
```

USER_ID	USERNAME	PASSWORD	U	A	FIRSTNAME	LASTNAME	GENDER	DOB	ADDR_STREET	ADDR_CITY	AD	ADDR_COUNTRY	PHONE	EMAIL
1	JamesBlunt	JamesBlunt	S	1	James	Blunt	Male	04-MAR-86	1234 Street	Baltimore	MD	USA	4436544567	JamesBlunt@gmail.com
2	JeffBlue	JeffBlue	S	1	Jeff	Blue	Male	08-JUN-89	1276 Street	Columbia	MD	USA	4436456567	JeffBlue@gmail.com
3	JackBlack	JackBlack	S	1	James	Black	Male	09-DEC-84	12776 Street	Baltimore	MD	USA	4436576987	JackBlack@gmail.com
4	AmyHack	AmyHack	S	1	Amy	Hack	Female	03-APR-86	127 Street	Ellicot	MD	USA	4436545467	AmyHack@gmail.com
5	EmaBlaze	EmaBlaze	S	1	Ema	Blaze	Female	07-JUL-89	1233 Street	Baltimore	MD	USA	4556856567	EmaBlaze@gmail.com
6	JesseJay	JesseJay	S	1	Jesse	Jay	Female	14-DEC-89	1212 Street	Towson	MD	USA	4490066987	JesseJay@gmail.com
7	RobertDowny	RobertDowny	D	1	Robert	Downy	Male	07-MAR-79	123A Belkin	Baltimore	MD	USA	4416676789	RobertD@gmail.com
8	RobertPattinson	RobertPattinson	D	1	Robert	Pattinson	Male	17-MAR-69	12342 Ave	Ellicot	MD	USA	3440986789	RobertPattinson@gmail.com
9	AceHarper	AceHarper	D	1	Ace	Harper	Male	27-JUN-72	123E Old	Towson	MD	USA	4410986569	AceHarper@gmail.com
10	AlisonGrey	AlisonGrey	D	1	Alison	Grey	Female	07-JAN-74	123WE West	Columbia	MD	USA	4414586789	AlisonGrey@gmail.com
11	MayHalley	MayHalley	D	1	May	Halley	Female	23-JUL-74	1231G Newdale	Baltimore	MD	USA	4789986789	MayHalley@gmail.com
12	KellyBrown	KellyBrown	D	1	Kelly	Brown	Female	13-MAY-79	123 Street	Halethorpe	MD	USA	4410086789	KellyBrown@gmail.com

```
select * from doctors;
```

S_ID	U	S_SALARY	HIRE_DATE	LEAVING_D	S	QUALIFICATION	WORK_EXP	S_TYPE	D_ID	M_ID
1	S	45000	05-JUN-14		1	Certification in sales	24	full_time	101	1
2	S	40000	02-MAY-14		1	Certification in sales	20	full_time	102	2
3	S	48000	25-NOV-13		1	Diploma in sales	29	full_time	103	3
4	S	45000	09-MAY-14		1	Certification in sales	24	full_time	104	4
5	S	40000	02-MAY-14		1	Certification in sales	20	full_time	105	5
6	S	40000	25-NOV-14		1	Certification in sales	20	full_time	106	6

DOC_ID	U	SPECIALITY	L	CLIENT_SI	S_ID
7	D	Allergy and Immunology	1	02-MAR-13	1
8	D	Dermatology	1	03-APR-13	2
9	D	Pediatrics	1	06-MAY-13	3
10	D	Family Practice	1	07-JUL-13	4
11	D	Otorhinolaryngology	1	09-SEP-13	5
12	D	Geriatric Medicine	1	04-APR-13	6

After procedure execution:

- **For Sales Rep**

exec insert_user('S', 'JohnYu', 'JohnYu', 'John', 'Yu', 'Male', date'1990-07-08', '1100 Westland BLVD', 'Baltimore', 'MD', 'USA', 4545454545, 'JohnYu@gmail.com', '45000', date'2014-11-12', '', 'Certification', 1, 'full_time', 104, 4, '', '');

exec insert_user('S', 'DexterMorgan', 'DexterMorgan', 'John', 'Morgan', 'Male', date'1980-09-08', '1100 Westland BLVD', 'Jacsonville', 'FL', 'USA', 4545454545, 'DexterMorgan@gmail.com', '45000', date'2014-11-12', '', 'Certification', 1, 'full_time', 104, 4, '', '');

- **For Doctor**

exec insert_user('D', 'JohnSnow', 'JohnSnow', 'John', 'Snow', 'Male', date '1990-07-08', '1100 Security BLVD', 'Catonsville', 'MD', 'USA', 4545774545, 'JohnSnow@gmail.com', '', '', '', '', '', '', '', '', 'Family Practice', 13);

select * from user_profile;
select * from sales_rep;
select * from doctors;

USER_ID	USERNAME	PASSWORD	U	A	FIRSTNAME	LASTNAME	GENDER	DOB	ADDR_STREET	ADDR_CITY	AD	ADDR_COUNTRY	PHONE	EMAIL
13	JohnYu	JohnYu	S	1	John	Yu	Male	08-JUL-90	1100 Westland BLVD	Baltimore	MD	USA	4545454545	JohnYu@gmail.com
14	JohnSnow	JohnSnow	D	1	John	Snow	Male	08-JUL-90	1100 Security BLVD	Catonsville	MD	USA	4545774545	JohnSnow@gmail.com
15	DexterMorgan	DexterMorgan	S	1	John	Morgan	Male	08-SEP-80	1100 Westland BLVD	Jacsonville	FL	USA	4545454545	DexterMorgan@gmail.com
1	JamesBlunt	JamesBlunt	S	1	James	Blunt	Male	04-MAR-86	1234 Street	Baltimore	MD	USA	4436544567	JamesBlunt@gmail.com
2	JeffBlue	JeffBlue	S	1	Jeff	Blue	Male	08-JUN-89	1276 Street	Columbia	MD	USA	4436456567	JeffBlue@gmail.com
3	JackBlack	JackBlack	S	1	James	Black	Male	09-DEC-84	12776 Street	Baltimore	MD	USA	4436576987	JackBlack@gmail.com
4	AmyHack	AmyHack	S	1	Amy	Hack	Female	03-APR-86	127 Street	Ellicot	MD	USA	4436545467	AmyHack@gmail.com
5	EmaBlaze	EmaBlaze	S	1	Ema	Blaze	Female	07-JUL-89	1233 Street	Baltimore	MD	USA	4556856567	EmaBlaze@gmail.com
6	JesseJay	JesseJay	S	1	Jesse	Jay	Female	14-DEC-89	1212 Street	Towson	MD	USA	4490066987	JesseJay@gmail.com
7	RobertDowny	RobertDowny	D	1	Robert	Downy	Male	07-MAR-79	123A Belkin	Baltimore	MD	USA	4416676789	RobertD@gmail.com
8	RobertPattinson	RobertPattinson	D	1	Robert	Pattinson	Male	17-MAR-69	12342 Ave	Ellicot	MD	USA	3440986789	RobertPattinson@gmail.com
9	AceHarper	AceHarper	D	1	Ace	Harper	Male	27-JUN-72	123E Old	Towson	MD	USA	4410986569	AceHarper@gmail.com
10	AlisonGrey	AlisonGrey	D	1	Alison	Grey	Female	07-JAN-74	123WE West	Columbia	MD	USA	4414586789	AlisonGrey@gmail.com
11	MayHalley	MayHalley	D	1	May	Halley	Female	23-JUL-74	1231G Newdale	Baltimore	MD	USA	4789986789	MayHalley@gmail.com
USER_ID	USERNAME	PASSWORD	U	A	FIRSTNAME	LASTNAME	GENDER	DOB	ADDR_STREET	ADDR_CITY	AD	ADDR_COUNTRY	PHONE	EMAIL
12	KellyBrown	KellyBrown	D	1	Kelly	Brown	Female	13-MAY-79	123 Street	Halethorpe	MD	USA	4410086789	KellyBrown@gmail.com

S_ID	U	S_SALARY	HIRE_DATE	LEAVING_D	S	QUALIFICATION	WORK_EXP	S_TYPE	D_ID	M_ID
13	S	45000	12-NOV-14		1	Certification	1	full_time	104	4
15	S	45000	12-NOV-14		1	Certification	1	full_time	104	4
1	S	45000	05-JUN-14		1	Certification in sales	24	full_time	101	1
2	S	40000	02-MAY-14		1	Certification in sales	20	full_time	102	2
3	S	48000	25-NOV-13		1	Diploma in sales	29	full_time	103	3
4	S	45000	09-MAY-14		1	Certification in sales	24	full_time	104	4
5	S	40000	02-MAY-14		1	Certification in sales	20	full_time	105	5
6	S	40000	25-NOV-14		1	Certification in sales	20	full_time	106	6

8 rows selected.

DOC_ID	U	SPECIALITY	L	CLIENT_SI	S_ID
14	D	Family Practice	1	29-NOV-15	13
7	D	Allergy and Immunology	1	02-MAR-13	1
8	D	Dermatology	1	03-APR-13	2
9	D	Pediatrics	1	06-MAY-13	3
10	D	Family Practice	1	07-JUL-13	4
11	D	Otorhinolaryngology	1	09-SEP-13	5
12	D	Geriatric Medicine	1	04-APR-13	6

3&16. The manager will assign tasks to the employees of his department. In order to complete the task, the sales rep can take appointments with the doctors and set up meetings, where the sales rep will talk about the new medicine and distribute samples.

Availability of a sales rep will be computed by using **start_datetime** and **end_datetime** of appointment for that day from **appointment** table, if sales rep doesn't have any appointment at that time then only new task can be assigned. [new task will be inserted in appointment table with status 1]

NOTE: Procedure insert_appointment will be used by manager to create a new task in appointment table. Status id will be 0 initially and will be updated by assigned sales rep to 1 after task is completed. Also expense and expense_desc will be updated by sales rep. Procedure update_task will be used by sales rep after completion of appointment.
Trigger sales_rep_schedule will check whether the appointment inserted in the table does collide with any other appointments on a day for the sales rep who is being assigned to the task(appointment).

Solution:-

Assumption: doctors can have appointment with 5 doctors at a time

```
create or replace TRIGGER  sales_rep_schedule
BEFORE insert or update of a_start_time ON appointment /*will be triggered before inserting record on appointment table and also before updating the column a_start_time.*/
for each row
declare
new_start_date timestamp;
new_end_date timestamp;
ct number;
begin
select count(*) into ct from appointment where s_id = :new.s_id and :new.a_start_time between a_start_time and a_end_time; /*will check for conflict of time and date*/
if ct > 0 then --if conflict found, throw error.
RAISE_APPLICATION_ERROR (-20000, 'Can not assign this task, sales rep has another task at that time.');
end if;
```

```
end;


create or replace procedure insert_appointment( s_time IN timestamp, e_time IN timestamp,
street IN VARCHAR, city IN VARCHAR, state IN VARCHAR, country IN VARCHAR,
description IN VARCHAR, sales_rep IN int, exp IN int, exp_desc IN VARCHAR, stat IN
CHAR,
doc_name1 IN varchar, doc_mail1 IN varchar, doc_name2 IN varchar, doc_mail2 IN varchar,
doc_name3 IN varchar, doc_mail3 IN varchar,
doc_name4 IN varchar, doc_mail4 IN varchar, doc_name5 IN varchar, doc_mail5 IN varchar)
IS
max_aid int;

d_id doctors.doc_id%type;
BEGIN
   SELECT count(*) into max_aid from appointment;
   INSERT into appointment values(max_aid + 1, s_time, e_time, street, city, state,country,
description, sales_rep, exp, exp_desc, stat);


   if doc_name1 IS NOT NULL and doc_mail1 IS NOT NULL then
      select distinct u.user_id into d_id from doctors d, USER_PROFILE u where d.s_id =
sales_rep and u.firstname=doc_name1 and u.email=doc_mail1; /*get doctor id(user_id) from
user profile for mapping by using email(unique)*/
      INSERT into doc_appointment values(max_aid + 1, d_id); /*mapping for appointment and
doctor table.*/
   end if;

   if doc_name2 IS NOT NULL and doc_mail2 IS NOT NULL then
      select distinct u.user_id into d_id from doctors d, USER_PROFILE u where d.s_id =
sales_rep and u.firstname=doc_name2 and u.email=doc_mail2; /*get doctor id(user_id) from
user profile for mapping by using email(unique)*/
```

```
    INSERT into doc_appointment values(max_aid + 1, d_id); --mapping for appointment and
doctor table.
  end if;

  if doc_name3 IS NOT NULL and doc_mail3 IS NOT NULL then
    select distinct u.user_id into d_id from doctors d, USER_PROFILE u where d.s_id =
sales_rep and u.firstname=doc_name3 and u.email=doc_mail3; /*get doctor id(user_id) from
user profile for mapping by using email(unique)*/
    INSERT into doc_appointment values(max_aid + 1, d_id); --mapping for appointment and
doctor table.
  end if;

  if doc_name4 IS NOT NULL and doc_mail4 IS NOT NULL then
    select distinct u.user_id into d_id from doctors d, USER_PROFILE u where d.s_id =
sales_rep and u.firstname=doc_name4 and u.email=doc_mail4; /*get doctor id(user_id) from
user profile for mapping by using email(unique)*/
    INSERT into doc_appointment values(max_aid + 1, d_id); --mapping for appointment and
doctor table.
  end if;

  if doc_name5 IS NOT NULL and doc_mail5 IS NOT NULL then
    select distinct u.user_id into d_id from doctors d, USER_PROFILE u where d.s_id =
sales_rep and u.firstname=doc_name1 and u.email=doc_mail1; /*get doctor id(user_id) from
user profile for mapping by using email(unique)*/
    INSERT into doc_appointment values(max_aid + 1, d_id); --mapping for appointment and
doctor table.
  end if;
END;
```

Procedure Execution(for insert_appointment) :

EXEC insert_appointment(timestamp '2015-12-01 14:30:00', timestamp '2015-12-01 15:30:00', '1124 West Street', 'Towson', 'MD', 'USA','Promotion of Carbinoxamine',7, 0, 'to be filled by Sales Rep', '0', 'May', 'MayHalley@gmail.com', 'Ace', 'AceSmith@gmail.com', 'Alison', 'AlisonLee@gmail.com',",",",",");

	A_ID	A_START_TIME	A_END_TIME	A_ADDR_STREET	A_ADDR_CITY	A_ADDR_STATE	A_ADDR_COUNTRY	A_DESC	S_ID	EXPENSE	EXPENSE_DESC	STATUS
1	25	01-DEC-15 02....	01-DEC-15 03....	1124 West S...	Towson	MD	USA	Promotion o...	7	0	to be fille...	0
2	1	11-NOV-15 01....	11-NOV-15 03....	123A Belkin	Baltimore	MD	USA	Promotion o...	1	280	travel=$80,...	0
3	2	12-NOV-15 10....	12-NOV-15 11....	12342 Ave	Ellicot	MD	USA	Promotion o...	2	180	travel=$30,...	0

	A_ID	DOC_ID
1	1	11
2	2	12
3	3	13
4	4	14
5	5	15
6	6	16
7	7	17
8	8	18
9	9	19
10	10	20
11	11	21
12	12	22
13	13	23
14	14	24
15	15	25
16	16	26
17	25	15
18	25	17
19	25	18
20	17	27
21	18	28
22	19	11
23	20	11
24	21	17
25	22	18

Execution for time duration the sales rep is occupied:

EXEC insert_appointment(timestamp '2015-12-02 10:00:00', timestamp '2015-12-02 11:30:00', '1124 West Street', 'Towson', 'MD', 'USA','Promotion of Brufen', 5, 0, 'to be filled by Sales Rep', '0', 'May', 'MayHalley@gmail.com', '', '', '', '','','','','');

```
Error starting at line : 1 in command -
EXEC insert_appointment(timestamp '2015-12-02 10:00:00', timestamp '2015-12-02 11:
Error report -
ORA-20000: Can not assign this task, sales rep has another task at that time.
ORA-06512: at "SYSTEM.SALES_REP_SCHEDULE", line 8
ORA-04088: error during execution of trigger 'SYSTEM.SALES_REP_SCHEDULE'
ORA-06512: at "SYSTEM.INSERT_APPOINTMENT", line 10
ORA-06512: at line 1
20000. 00000 -  "%s"
*Cause:    The stored procedure 'raise_application_error'
           was called which causes this error to be generated.
*Action:   Correct the problem as described in the error message or contact
           the application administrator or DBA for more information.
```

Before Procedure Execution(for update_appointment):

select * from appointment where a_id = 5;

select * from distribution where a_id = 5;

A_ID	A_START_TIME	A_END_TIME	A_ADDR_STREET	A_ADDR_CITY	A_	A_ADDR_COUNTRY	A_DESC	S_ID	EXPENSE	EXPENSE_DESC	S
5	12-NOV-15 10.00.00.000000 AM	12-NOV-15 11.30.00.000000 AM	226 Street	Baltimore	MD	USA	Promotion of Brufen	5	3500	travel=$2000, hotel and food=$1500	0

no rows selected

After Procedure Execution (for update_appointment):

EXEC update_appointment(5, 3000, 'travel=$2000, hotel and food=$1000', 'Product Distributed', 'Carbinoxamine',10);

select * from appointment where a_id = 5;

select * from distribution where a_id = 5;

A_ID	A_START_TIME	A_END_TIME	A_ADDR_STREET	A_ADDR_CITY	A_	A_ADDR_COUNTRY	A_DESC	S_ID	EXPENSE	EXPENSE_DESC	S
5	12-NOV-15 10.00.00.000000 AM	30-NOV-15 02.35.17.037000 PM	226 Street	Baltimore	MD	USA	Promotion of Brufen *** Product Distributed	5	3000	travel=$2000, hotel and food=$1000	1

TRANS_ID	TRANS_DAT	QUANTITY	P_ID	A_ID
11	30-NOV-15	10	1	5

---

4. Managers will insert new records of product(p_id, p_name, p_man_date, p_exp_date, ingredients, dosage, warning, status)

Also manager will modify some data of product such as (dosage, warning, status). They can also delete records (Delete From product where pname = ‘user given value’;)

Solution:-

```
CREATE OR REPLACE PROCEDURE insert_product(product_name IN varchar, man_date IN date, exp_date IN date, ing IN varchar, dos IN varchar, war IN varchar, stat IN varchar)
IS
max_pid int;
BEGIN
select count(*) into max_pid from product;
INSERT into product values(max_pid+1,product_name, man_date, exp_date, ing, dos, war, stat);
END;
```

For delete:

```
DELETE FROM product where p_id = &product_id;
```

For update:

```
UPDATE product SET dosage = '&dosage', warning = '&warning', status = &status WHERE p_id = &product_id;
```

Before Procedure Execution:

```
select * from product;
```

P_ID	P_NAME	P_MAN_DAT	P_EXP_DAT	INGREDIENTS	DOSAGE	WARNING	STATUS
1	Carbinoxamine	03-JUN-15	03-JUN-17	ingredients	1 or 2 tablets (4 to 8 mg) 3 to 4 times daily	Caution should be exercised in patients with history of asthma, glaucoma, difficulty in urinating, stomach or bladder obstruction, prostate problems	1
2	Cetrizine	04-JUL-15	04-JUL-16	ingredients	The recommended dose is 10 mg once daily	Caution should be exercised in patients with history of kidney or liver	1
3	Dimethindene	09-APR-15	09-APR-17	ingredients	1-2 mg 3 times-day. Topical As 0.1% gel-lotion: Use as directed	Caution should be exercised in patients with history of elderly, increased eye pressure, prostate enlargement, epilepsy	1
4	Ebastine	14-NOV-15	14-NOV-18	ingredients	The recommended dose is 10-20 mg once daily	Caution should be exercised in patients with history of liver and kidney impairment, QT interval prolongation, during pregnancy and breastfeeding	1
5	Ketotifin	24-AUG-15	24-AUG-17	ingredients	1 mg twice daily, up to 2 mg twice daily if needed	It may cause dizziness or drowsiness, do not drive a car or operate machinery while taking this medication	1
6	Methdilazine	15-MAR-15	15-MAR-16	ingredients	8-16 mg twice daily	It may cause drowsiness or blurred vision, do not drive a car or operate machinery while taking this medication	1

After Procedure Execution:

EXEC insert_product('Aspirin',date '2015-6-3',date '2017-6-3', 'ingredients', '1 or 2 tablets (4 to 8 mg) 3 to 4 times daily','Caution should be exercised in patients with history of asthma, glaucoma, difficulty in urinating, stomach or bladder obstruction, prostate problems',1);

ID	P_NAME	P_MAN_DAT	P_EXP_DAT	INGREDIENTS	DOSAGE	WARNING	STATUS
7	Aspirin	03-JUN-15	03-JUN-17	ingredients	1 or 2 tablets (4 to 8 mg) 3 to 4 times daily	Caution should be exercised in patients with history of asthma, glaucoma, difficulty in urinating, stomach or bladder obstruction, prostate problems	1
1	Carbinoxamine	03-JUN-15	03-JUN-17	ingredients	1 or 2 tablets (4 to 8 mg) 3 to 4 times daily	Caution should be exercised in patients with history of asthma, glaucoma, difficulty in urinating, stomach or bladder obstruction, prostate problems	1
2	Cetrizine	04-JUL-15	04-JUL-16	ingredients	The recommended dose is 10 mg once daily	Caution should be exercised in patients with history of kidney or liver	1
3	Dimethindene	09-APR-15	09-APR-17	ingredients	1-2 mg 3 times-day. Topical As 0.1% gel-lotion: Use as directed	Caution should be exercised in patients with history of elderly, increased eye pressure, prostate enlargement, epilepsy	1
4	Ebastine	14-NOV-15	14-NOV-18	ingredients	The recommended dose is 10-20 mg once daily	Caution should be exercised in patients with history of liver and kidney impairment, QT interval prolongation, during pregnancy and breastfeeding	1
5	Ketotifin	24-AUG-15	24-AUG-17	ingredients	1 mg twice daily, up to 2 mg twice daily if needed	It may cause dizziness or drowsiness, do not drive a car or operate machinery while taking this medication	1
6	Methdilazine	15-MAR-15	15-MAR-16	ingredients	8-16 mg twice daily	It may cause drowsiness or blurred vision, do not drive a car or operate machinery while taking this medication	1

select * from product;

5 & 6. Once sales representative has completed task then s/he will update appointment status, appointment end date time, expense, expense description, number of samples distributed(a_status

= ‘1’ for completed appointment, a_end_datetime, expense, expense_desc, quantity) and append any comment about the appointment to appointment description.

ID of the product, quantity of that product and date of distribution will be inserted for each product distributed to doctors by the sales representative.

The sales rep can also append any comment about the appointment

***NOTE1 :** update query will run first and insert the quantity of product distributed, if more than one product was distributed then sales rep can use the same execution query only changing the product name and quantity. If condition will check the values for appointment table field and if it is same then it won’t run the update and directly run insert query for distribution table.

Solution:-

```
create or replace procedure update_appointment(app_id IN number, exp IN int, exp_desc IN
varchar, app_desc IN VARCHAR2, prod IN varchar2, quant IN number)
IS
pid PRODUCT.P_ID%TYPE;
tid distribution.trans_id%type;
a_expense int;
a_expense_desc varchar(255);
a_status char(1);
appdesc varchar2(255);
newdesc varchar2(255);
BEGIN
  select a_desc, expense, expense_desc, status into appdesc, a_expense, a_expense_desc,a_status
from APPOINTMENT where a_id = app_id;
  newdesc := appdesc || ' *** ' || app_desc; /*append new description to the older one.*/
if (a_expense != exp and a_expense_desc != exp_desc and a_status != 1) then -- *NOTE1
  UPDATE appointment SET a_end_time = systimestamp ,expense = exp , expense_desc =
exp_desc , a_desc = newdesc, status = 1  where a_id = app_id;
dbms_output.put_line('Appointment table updated for appointment id ' || app_id);
end if;
  select p_id into pid from product where p_name = prod;
  select count(*) into tid from Distribution;
```

INSERT into Distribution values(tid+1,sysdate,quant,pid,app_id); --to keep record of samples distributed./*to keep record of samples distributed.*/

END;

Before Procedure Execution:

select * from appointment where a_id = 5;

select * from distribution where a_id = 5;

A_ID	A_START_TIME	A_END_TIME	A_ADDR_STREET	A_ADDR_CITY	A_	A_ADDR_COUNTRY	A_DESC	S_ID	EXPENSE	EXPENSE_DESC	S
5	12-NOV-15 10.00.00.000000 AM	12-NOV-15 11.30.00.000000 AM	226 Street	Baltimore	MD	USA	Promotion of Brufen	5	3500	travel=$2000, hotel and food=$1500	0

no rows selected

After Procedure Execution:

EXEC update_appointment(5, 3000, 'travel=$2000, hotel and food=$1000', 'Product Distributed', 'Carbinoxamine',10);

select * from appointment where a_id = 5;

select * from distribution where a_id = 5;

A_ID	A_START_TIME	A_END_TIME	A_ADDR_STREET	A_ADDR_CITY	A_	A_ADDR_COUNTRY	A_DESC	S_ID	EXPENSE	EXPENSE_DESC	S
5	12-NOV-15 10.00.00.000000 AM	30-NOV-15 02.35.17.037000 PM	226 Street	Baltimore	MD	USA	Promotion of Brufen *** Product Distributed	5	3000	travel=$2000, hotel and food=$1000	1

TRANS_ID	TRANS_DAT	QUANTITY	P_ID	A_ID
11	30-NOV-15	10	1	5

For another product distribution:

EXEC update_appointment(5, 3000, 'travel=$2000, hotel and food=$1000', 'Product Distributed', 'Aspirin',20);

select * from appointment where a_id = 5;

select * from distribution where a_id = 5;

A_ID	A_START_TIME	A_END_TIME	A_ADDR_STREET	A_ADDR_CITY	A_	A_ADDR_COUNTRY	A_DESC	S_ID	EXPENSE	EXPENSE_DESC	S
5	12-NOV-15 10.00.00.000000 AM	30-NOV-15 02.35.17.037000 PM	226 Street	Baltimore	MD	USA	Promotion of Brufen *** Product Distributed	5	3000	travel=$2000, hotel and food=$1000	1

TRANS_ID	TRANS_DAT	QUANTITY	P_ID	A_ID
12	30-NOV-15	20	7	5
11	30-NOV-15	10	1	5

---

8. In case if sales rep leaves or gets salary raise, then columns like salary, leaving_date and status will be updated. [status will be changed to 0, 0 : no longer the employee of the company]

NOTE: rep_leaves procedure checks whether all the appointments are completed before leaving the company.

Solution:-

If sales rep leaves:

CREATE or REPLACE PROCEDURE rep_leaves(rep_id in number)

as

```
cursor c1 is select a_end_time from appointment where s_id=rep_id and status=1; /*gets the end date and time for appointments to check whether any appointment is pending before leaving*/
last_date timestamp;
BEGIN
open c1;
loop
fetch c1 into last_date;
exit when c1%notfound;
if (to_char(last_date,'YYYY-MM-DD') < to_char(sysdate,'YYYY-MM-DD')) then
update sales_rep set leaving_date=sysdate, status=0 where s_id=rep_id;
else
dbms_output.put_line('Please contact manager before deleting the employee. Some appointments might be pending.'); /*deleting employing means making their status 0*/
end if;
end loop;
close c1;
END;
```

Before Procedure Execution:

```
select * from sales_rep where s_id = 1;
```

S_ID	U	S_SALARY	HIRE_DATE	LEAVING_D	S	QUALIFICATION	WORK_EXP	S_TYPE	D_ID	M_ID
1	S	45000	05-JUN-14		1	Certification in sales	24	full_time	101	1

After Procedure Execution:

```
exec rep_leaves(1);
select * from sales_rep where s_id = 1;
```

PL/SQL procedure successfully completed.

S_ID	U	S_SALARY	HIRE_DATE	LEAVING_D	S	QUALIFICATION	WORK_EXP	S_TYPE	D_ID	M_ID
1	S	45000	05-JUN-14	30-NOV-15	0	Certification in sales	24	full_time	101	1

For employee with uncompleted task:

ID	A_START_TIME	A_END_TIME	A_ADDR_STREET	A_ADDR_CITY	A_	A_ADDR_COUNTRY	A_DESC	S_ID	EXPENSE	EXPENSE_DESC	S
8	01-DEC-15 02.30.00.000000 PM	02-DEC-15 11.55.57.971000 PM	1124 West Street	Towson	MD	USA	Promotion of Carbinoxamine *** Product Distributed	13	300	travel=$200, hotel and food=$100	1

select * from appointment where s_id = 13;

```
exec rep_leaves(13);
```

Execute | Save Script | Clear Screen | Cancel

Please contact manager before deleting the employee. Some appointments might be pending.
PL/SQL procedure successfully completed.

If sales rep gets salary raise:

```
CREATE or REPLACE PROCEDURE raise_salary(rep_id in integer, amount in decimal)
is
stat char(1);
fraction number;
begin
 select status into stat from SALES_REP where s_id=rep_id;
 fraction := 1 + (amount/100); --amount to be raised is in percent
 If stat = 1 then --checks for a valid sales rep
  update sales_rep set s_salary = s_salary * fraction where s_id=rep_id;
  else
```

```
   dbms_output.put_line('Sales rep is not active.');
  End if;
exception
 when no_data_found then
  dbms_output.put_line('no such employee found');
end;
```

Before Procedure Execution:

```
select * from sales_rep where s_id = 13;
```

S_ID	U	S_SALARY	HIRE_DATE	LEAVING_D	S	QUALIFICATION	WORK_EXP	S_TYPE	D_ID	M_ID
13	S	45000	12-NOV-14		1	Certification	1	full_time	104	4

After Procedure Execution:

```
exec raise_salary(13,20);
select * from sales_rep where s_id = 13;
```

S_ID	U	S_SALARY	HIRE_DATE	LEAVING_D	S	QUALIFICATION	WORK_EXP	S_TYPE	D_ID	M_ID
13	S	54000	12-NOV-14		1	Certification	1	full_time	104	4

9.a. Sales representatives will be ranked based up on the quantity of sample distribution, for which ‘quantity’ of distribution will be computed from ‘distribution’ table.

9.b Sales representatives will also be ranked based on the no. of new clients, for which count of 'client_since_date' in 'doctors' table will give number of new doctors and 's_id' will give ID of sales representative who brought in the doctor.

9.c Sales representatives will also be ranked based up on the feedback given by doctors after appointment which will be calculated using 'performance_level' from fb_sales_rep table.

An anonymous procedures calls for the procedure rank_sales_rep. Based up on the input, performance is either calculated for samples distributed, clients gained or by feedbacks from doctors.

Solution:-

```
CREATE or REPLACE FUNCTION calculate_performance(sales_id IN number)
return decimal
is
/*Feedbacks are given as great, good, average and bad which are represented in the range of 4 to 1 (4 = great and 1 = bad). This function converts each feedback in points which can be used for ranking sales reps. The function returns total points for each sales rep. This function is called in procedure rank_sales_rep.*/
cursor c1 is select count(*) as cnt,performance_level,s_id from FB_SALES_REP where s_id=sales_id group by s_id,performance_level;
per_level varchar2(225);
total number;
quant number;
sal_id number;
begin
total := 0;
open c1;
loop
fetch c1 into quant,per_level,sal_id;
```

```
exit when c1%notfound;
if (per_level = 'great') then
   total := total + 4*quant;
elsif (per_level = 'good') then
   total := total + 3*quant;
elsif (per_level = 'average') then
   total := total + 2*quant;
elsif (per_level = 'bad') then
   total := total + 1*quant;
end if;
end loop;
close c1;
return total; /*returns cumulated feedback for every sales rep*/
end;

/*This function will assign benefits to sales rep*/
create or replace FUNCTION assign_benefits(sales_id in NUMBER)
RETURN number
is
bendesc benefit.ben_desc%type;
benid benefit_map.ben_id%type;
max_benid BENEFIT.BEN_ID%TYPE;
 BEGIN
 select max(ben_id) into max_benid from BENEFIT;
select max(ben_id) into benid from BENEFIT_MAP where S_ID = sales_id;
  benid := benid + 1;
  if benid > max_benid then /* If sales rep have all benefits then no more benefits are assigned.*/
    dbms_output.put_line('Sales rep ' || sales_id || ' already have all benefits.');
          return 0;
  else /* Assign benefit to sales rep. */
  insert into BENEFIT_MAP values(sales_id, sysdate, benid);
```

```
  select ben_desc into bendesc from BENEFIT b where b.BEN_ID = benid;
  dbms_output.put_line('Benifit: '|| bendesc || ' ' ||  'is given to sales rep id: '||sales_id);
  RETURN 1;
  end if;
end;


CREATE or REPLACE PROCEDURE rank_sales_rep(var1 IN varchar2)
is
/*c1 is used to get the quantity of samples distributed by each sales rep in a period of 6 months. Based on this, the sales rep having distribution quantity greater than 40% of the average will be qualified for benefits.*/
cursor c1 is select sum(d.quantity) as quantity_distributed, a.s_id from appointment a, distribution d, sales_rep s where a.a_id = d.a_id and s.s_id = a.s_id and s.hire_date <= sysdate - interval '6' month group by a.s_id ORDER BY quantity_distributed desc;
quant number;
sales_id sales_rep.s_id%type;
per_level varchar(225);
total number;

/*c3 is used to get the number of new clients by each sales rep in a period of 6 months. Based on this, the sales rep having new clients greater than average will be qualified for benefits.*/

cursor c3 is select count(*) as new_clients, d.s_id from doctors d, sales_rep s where d.s_id = s.s_id and d.client_since_date >= sysdate - interval '1' year and  s.hire_date <= sysdate - interval '6' month group by d.s_id ORDER BY new_clients;
temp number;
temp1 number;
temp2 number;
i number;

a_benefit number;
```

```
/*c5 is used to get the sales rep id from the feedback table which has been passed as an input parameter for the function calculate_performance. Each sales rep having feedback points greater than 40% of the average will be qualified for benefits.*/

cursor c5 is select distinct f.s_id from fb_sales_rep f, sales_rep s where s.s_id = f.s_id and s.hire_date <= sysdate - interval '1' year ORDER BY s_id asc;

type NumberArray is table of number index by binary_integer;
myArray NumberArray;

begin
total :=0;
temp :=0;
temp1:=0;
temp2:=0;
i:=0;

/*for calculating performance of sales rep based on number of distributed samples.*/
if (var1 = 'distribution_quantity') then
dbms_output.put_line('rank_distribution_quantity for sales rep');
/*for fetching data in array*/
open c1;
 loop
  fetch c1 into quant, sales_id;
  exit when c1%notfound;
  myarray(sales_id) := quant;
  i:=i+myarray(sales_id); /*Total for calculating average.*/
 end loop;
close c1;
temp:=i/myarray.count;
temp2:=temp*1.4; /*criteria of 40% greater than average*/
```

```
/*for displaying filtered data from array*/
open c1;
 loop
  fetch c1 into quant, sales_id;
  exit when c1%notfound;
  if(myarray(sales_id)>temp2)then
    dbms_output.put_line('Quantity: '|| myarray(sales_id) || ' ' ||  'by sales rep id: '||sales_id);
          a_benefit := ASSIGN_BENEFITS(sales_id);
  end if;
 end loop;
close c1;

/*for calculating performance of sales rep based on number of new clients in past 6 months.*/
elsif (var1 = 'new_client') then
  dbms_output.put_line('rank_new_client for sales rep hired one year before');
/*for fetching data in array*/
open c3;
 loop
  fetch c3 into quant, sales_id;
  exit when c3%notfound;
  myarray(sales_id) := quant;
  i:=i+myarray(sales_id);
 end loop;
close c3;
 temp:=i/myarray.count;
/*for displaying filtered data from array*/
open c3;
 loop
  fetch c3 into quant, sales_id;
  exit when c3%notfound;
  if(myarray(sales_id)>temp)then
```

```
   dbms_output.put_line('Number of new clients: '|| myarray(sales_id) || ' ' ||  'by sales rep id:
'||sales_id);
          a_benefit := ASSIGN_BENEFITS(sales_id);
  end if;
 end loop;
close c3;

/*for calculating performance of sales rep based on the feedback given by doctors.*/
elsif (var1 = 'feedback') then
  dbms_output.put_line('Performance for Feedback');
/*for fetching data in array*/
  open c5;
  loop
  fetch c5 into sales_id;
  exit when c5%notfound;
          myarray(sales_id) := calculate_performance(sales_id);
    i:=i+myarray(sales_id);
  end loop;
  temp1:=myarray.count;
  temp2:=i/temp1;
  temp:=temp2*1.4;
  close c5;
/*for displaying filtered data from array*/
  open c5;
  loop
  fetch c5 into sales_id;
  exit when c5%notfound;
          if (myarray(sales_id) > temp) then
     dbms_output.put_line('Sales Rep ID: '||sales_id ||' = ' || myarray(sales_id));
          a_benefit := ASSIGN_BENEFITS(sales_id);
          end if;
```

```
  end loop;
  close c5;
else
  dbms_output.put_line('Error');
end if;
end;

DECLARE
rank1 varchar2(255);
rank2 varchar2(255);
rank3 varchar2(255);
begin
rank1 :='distribution_quantity';
rank2 :='new_client';
rank3 :='feedback';
rank_sales_rep(rank1);
rank_sales_rep(rank2);
rank_sales_rep(rank3);
END;
```

```
PL/SQL procedure successfully completed.

rank_distribution_quantity for sales rep
Quantity: 90 by sales rep id: 8
Benifit: These may include tuition assistance, flexible medical or child-care spending accounts is given to sales rep id: 8
Quantity: 85 by sales rep id: 7
Benifit: These may include tuition assistance, flexible medical or child-care spending accounts is given to sales rep id: 7

PL/SQL procedure successfully completed.

rank_new_client for sales rep hired one year before
Number of new clients: 2 by sales rep id: 10
Benifit: These may include tuition assistance, flexible medical or child-care spending accounts is given to sales rep id: 10

PL/SQL procedure successfully completed.

Performance for Feedback
Sales Rep ID: 7 = 19
Benifit: These may include tuition assistance, flexible medical or child-care spending accounts is given to sales rep id: 7
Sales Rep ID: 10 = 11
Benifit: Paid time off are holidays, sick leave, and vacation leave. is given to sales rep id: 10
```

	S_ID	BEN_DATE	BEN_ID
1	1	05-JUN-14	1
2	2	02-MAY-14	1
3	3	25-NOV-13	1
4	4	09-MAY-14	1
5	5	02-MAY-14	1
6	7	01-DEC-15	3
7	8	01-DEC-15	3
8	7	01-DEC-15	4
9	10	01-DEC-15	3
10	7	01-DEC-15	5
11	10	01-DEC-15	4
12	6	25-NOV-14	1
13	7	20-JUN-14	1
14	8	20-MAY-14	1
15	10	09-APR-14	1
16	1	05-JUN-14	2
17	2	02-MAY-14	2
18	3	25-NOV-13	2
19	4	09-MAY-14	2
20	5	02-MAY-14	2
21	6	25-NOV-14	2
22	7	20-JUN-14	2
23	8	20-MAY-14	2
24	9	01-DEC-13	2
25	10	09-APR-14	2

---

11. Number of sales by sales representatives of each department will also be calculated for which d_id(department id) in ‘sales_rep’ table and ‘quantity’ of distribution of sample by that sales reps in distribution table will be used.

Solution:-

SELECT de.d_id, p.p_name, sum(di.quantity) as Total
from PRODUCT p, DISTRIBUTION di, DEPARTMENT de, SALES_REP sr, APPOINTMENT ap where de.d_id = sr.d_id and sr.s_id = ap.s_id and ap.a_id = di.a_id and di.p_id = p.p_id
Group by de.d_id, p.p_name;

D_ID	P_NAME	TOTAL
106	Carbinoxamine	10
102	Cetrizine	10
103	Dimethindene	10
104	Ebastine	15
101	Carbinoxamine	20

---

12. Budget will be estimated from appointment table using average of expense column for a particular city.

Solution:-

SELECT to_char(avg(expense),'fm9999999.99') as budget, a_addr_city from appointment where status = 1 group by a_addr_city where status = 1;

BUDGET	A_ADDR_CITY
4000.	Columbia
4500.	Laurel
5000.	Elkridge
3000.	Towson
300.	Frederick
3000.	Catonsville

13. Performance level of each sales rep will be computed based on no. of sales, feedback of performance level, no of new clients as mentioned above in (9) for each quarter and each sale reps are expected to do better than previous quarter.

Solution:-

CREATE OR REPLACE PROCEDURE perf_by_quarter(sid IN int)

```
IS
distribution_quantity int;
lastq_distribution_quantity  int;
new_clients int;
lastq_new_clients int;
great_perf int;
lastq_great_perf int;
BEGIN
/*for calculating the distribution of current and previous quarters */
select sum(d.quantity) into distribution_quantity  from distribution d, appointment a  WHERE a.a_id = d.a_id  and trans_date  >= sysdate - interval '3' month and a.s_id = sid;
select sum(d.quantity)  into lastq_distribution_quantity from distribution d, appointment a WHERE a.a_id = d.a_id  and trans_date  BETWEEN sysdate - interval '6' month and sysdate - interval '3' month and a.s_id = sid;
/*calculating the number of new clients by the sales rep for current and previous quarters */
select count(*) into new_clients from doctors  where client_since_date >= sysdate - interval '3' month and s_id = sid;
select count(*) into lastq_new_clients from doctors  where client_since_date BETWEEN sysdate - interval '6' month and sysdate - interval '3' month and s_id = sid;
/*calculating the number great feedbacks received by the sales rep in the current and previous quarters */
select count(*) into great_perf from fb_sales_rep fs, feedback f where fs.f_id =f.f_id and fs. performance_level = 'great' and f.f_datetime >= sysdate - interval '3' month and fs.s_id = sid;
select count(*) into lastq_great_perf from fb_sales_rep fs, feedback f where fs.f_id =f.f_id and fs. performance_level = 'great' and f.f_datetime BETWEEN sysdate - interval '6' month and sysdate - interval '3' month and fs.s_id = sid;

/*comparing above calulation for current and previous quarter */
if distribution_quantity < lastq_distribution_quantity  then
dbms_output.put_line('Previous quarter: '||lastq_distribution_quantity  ||' This quarter: '|| distribution_quantity ||' Distribution rate of the employee is lagging!!');
```

```
else
dbms_output.put_line('Previous quarter: '||lastq_distribution_quantity ||' This quarter: '|| distribution_quantity ||' Distribution rate by the sales rep looks pretty good.');
end if;
if new_clients < lastq_new_clients then
dbms_output.put_line('Previous quarter: '||lastq_new_clients||' This quarter: '|| new_clients ||' Number of new clients are lower than previous quarter!!');
else
dbms_output.put_line('Previous quarter: '||lastq_new_clients||' This quarter: '|| new_clients ||' Number of new clients by the sales rep is going up, Good!');
end if;
if great_perf < lastq_great_perf then
dbms_output.put_line('Previous quarter: '||lastq_great_perf||' This quarter: '|| great_perf ||' Feedbacks aren't looking as good this quarter!!');
else
dbms_output.put_line('Previous quarter: '||lastq_great_perf||' This quarter: '|| great_perf ||' The employee is getting pretty good feedbacks.');
end if;
END;


select a.a_id, quantity, s_id, d.p_id, trans_date from distribution d , appointment a where a.a_id = d.a_id and a.s_id = 13;
select * from doctors where s_id = 13;
select * from fb_sales_rep where s_id = 13;
EXEC perf_by_quarter(13);
```

OUTPUT:

A_ID	QUANTITY	S_ID	P_ID	TRANS_DAT
7	10	13	1	30-NOV-15
7	10	13	4	30-JUL-15
8	15	13	1	30-NOV-15
9	25	13	1	30-NOV-15
10	35	13	1	30-NOV-15

DOC_ID	U	SPECIALITY	L	CLIENT_SI	S_ID
16	D	Family Practice	1	30-NOV-15	13
17	D	physiotherapy	1	30-NOV-15	13
18	D	physiotherapy	1	30-NOV-15	13
14	D	Family Practice	1	29-NOV-15	13

F_ID	F	PERFORMANCE_LEVEL	S_ID
13	S	great	13
14	S	great	13
15	S	great	13

Previous quarter: 10 This quarter: 85 Distribution rate by the sales rep looks pretty good.
Previous quarter: 0 This quarter: 4 Number of new clients by the sales rep is going up, Good!
Previous quarter: 0 This quarter: 3 The employee is getting pretty good feedbacks.
PL/SQL procedure successfully completed.

---

14. Quantity of distribution for each product will be computed for each location using quantity column from distribution table and location(city or state) from appointment table

Solution:-

```
SELECT a.a_addr_city as CITY, p.p_name as PRODUCT, SUM(d.quantity) as
Distributed_Quantity
from distribution d, appointment a, product p
where d.a_id=a.a_id and d.p_id=p.p_id
group by a.a_addr_city, p.p_name;
```

CITY	PRODUCT	DISTRIBUTED_QUANTITY
Frederick	Carbinoxamine	10
Catonsville	Ebastine	15
Towson	Carbinoxamine	95
Columbia	Carbinoxamine	20
Elkridge	Cetrizine	10
Laurel	Dimethindene	10

---

15. Doctors and/or sales representatives will be notified of expiry date before three days for a product by using expiry_date from product table and using interval.

Solution:-

/*To automate the execution of procedure, we have added an additional feature. The job class from scheduler package will make sure the procedure check_expiry ***is executed daily.*/***

```
BEGIN

  DBMS_SCHEDULER.CREATE_JOB (
          job_name => '"SYSTEM"."CHECK_EXPIRY_JOB"',
          job_type => 'STORED_PROCEDURE',
          job_action => 'SYSTEM.CHECK_EXPIRY',
          number_of_arguments => 0,
          start_date => TO_TIMESTAMP_TZ('2015-11-21 15:56:42.000000000
AMERICA/NEW_YORK','YYYY-MM-DD HH24:MI:SS.FF TZR'),
          repeat_interval => 'FREQ=DAILY',
          end_date => NULL,
          enabled => FALSE,
          auto_drop => FALSE,
          comments => 'check expiry date of products');

          DBMS_SCHEDULER.enable(
           name => '"SYSTEM"."CHECK_EXPIRY_JOB"');
END;
```

Before:

P_ID	P_NAME	P_MAN_DATE	P_EXP_DATE	INGREDIENTS	DOSAGE	WARNING	STATUS
1	Carbinoxamine	03-JUN-15	03-JUN-17	ingredients	1 or 2 ...	Caution shoul...	1
2	Cetrizine	04-JUL-15	04-JUL-16	ingredients	The rec...	Caution shoul...	1
3	Dimethindene	09-APR-15	09-APR-17	ingredients	1-2 mg ...	Caution shoul...	1
4	Ebastine	14-NOV-15	14-NOV-18	ingredients	The rec...	Caution shoul...	1
5	Ketotifin	24-AUG-15	24-AUG-17	ingredients	1 mg tw...	It may cause ...	1
6	Methdilazine	15-MAR-15	29-NOV-15	ingredients	8-16 mg...	It may cause ...	1
7	Aspirin	03-JUN-15	28-NOV-15	ingredients	1 or 2 ...	Caution shoul...	0

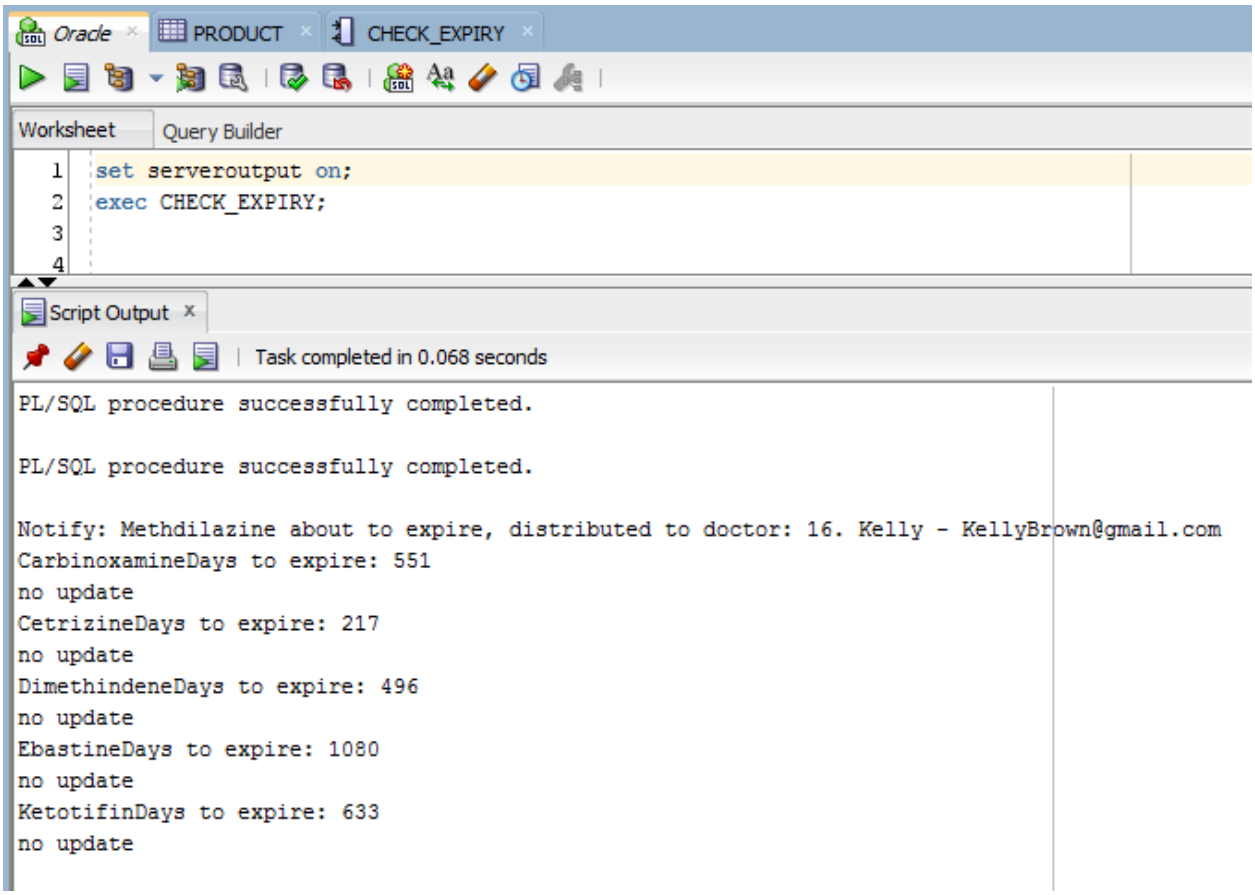
After:

P_ID	P_NAME	P_MAN_DATE	P_EXP_DATE	INGREDIENTS	DOSAGE	WARNING	STATUS
1	Carbinoxamine	03-JUN-15	03-JUN-17	ingredients	1 or 2 ...	Caution shoul...	1
2	Cetrizine	04-JUL-15	04-JUL-16	ingredients	The rec...	Caution shoul...	1
3	Dimethindene	09-APR-15	09-APR-17	ingredients	1-2 mg ...	Caution shoul...	1
4	Ebastine	14-NOV-15	14-NOV-18	ingredients	The rec...	Caution shoul...	1
5	Ketotifin	24-AUG-15	24-AUG-17	ingredients	1 mg tw...	It may cause ...	1
6	Methdilazine	15-MAR-15	29-NOV-15	ingredients	8-16 mg...	It may cause ...	0
7	Aspirin	03-JUN-15	28-NOV-15	ingredients	1 or 2 ...	Caution shoul...	0

/*This procedure will check for the products which are about to get expired. It will notify the sales rep and doctor if the expiry date is near(for 3 days and less than three days). It will also update the status of expired product as 0. */

```
create or replace procedure check_expiry
is
/*c1 gets the expiry dates of the products distributed to the doctors.*/
cursor c1 is select distinct p.p_exp_date, p.p_name, da.doc_id, u.FIRSTNAME, u.EMAIL
from product p, doc_appointment da, distribution d, appointment a, doctors doc,
USER_PROFILE u
where p.p_id=d.p_id and d.a_id = a.a_id and da.a_id = a.a_id and da.doc_id = doc.doc_id and
doc.doc_id = u.USER_ID;

/*c2 gets all the active products whose expiry dates are checked.*/
cursor c2 is select p_id, p_exp_date, p_name from product where status = 1;

expiry_date PRODUCT.P_EXP_DATE%TYPE;
product_name PRODUCT.P_NAME%TYPE;
doctor_id DOCTORS.DOC_ID%type;
diff number;
product_id PRODUCT.P_ID%TYPE;
fname USER_PROFILE.FIRSTNAME%TYPE;
mail USER_PROFILE.EMAIL%TYPE;

begin
/*to notify if product about to expire*/
open c1;
loop
fetch c1 into expiry_date, product_name, doctor_id, fname, mail;

diff := trunc(expiry_date) - trunc(sysdate) ; /*calculate the difference of dates in days*/
if diff <= 3 then
 dbms_output.put_line('Notify: ' || product_name ||' about to expire, distributed to doctor: '||
doctor_id || '. ' || fname || ' - '|| mail);
end if;
```

```
exit when c1%notfound;
end loop;
close c1;
/*update status to 0 if product expired.*/
open c2;
 loop
 fetch c2 into product_id, expiry_date, product_name;
 exit when c2%notfound;
 diff := trunc(expiry_date) - trunc(sysdate); /*calculate the difference of dates in days*
 dbms_output.put_line(product_name || 'Days to expire: '||diff);
 if (diff > 0) then
  dbms_output.put_line('no update');
 elsif (diff <= 0) then
  update product
  set status = 0
  where p_id = product_id;
  dbms_output.put_line('status updated for ' || product_name || 'having ' || expiry_date);
 else
  dbms_output.put_line('error');
 end if;
 end loop;
 close c2;
end;
```



---

17. Rating of samples will be computed using average of all ratings given by doctors using rate column from fb_product table.

Solution:-

```
CREATE or REPLACE PROCEDURE product_rate(pid IN int)
IS
product_rate INTEGER;
BEGIN
select avg(rate) into product_rate from fb_product where  p_id = pid;
dbms_output.put_line('Rate of the product is: '|| product_rate);
if product_rate < 2 then
dbms_output.put_line('Product should be sent to lab for evaluation ');
```

```
elsif product_rate > 4 then
dbms_output.put_line('Product is recommended for permanent manufacturing ');
else
dbms_output.put_line('Product rate is satisfactory ');
end if;
END;
```

OUTPUT:

```
select * from fb_product where p_id = 1;
exec PRODUCT_RATE(1);
```

F_ID	F	RATE	SIDE_EFFECT	P_ID
1	P	4	Dizziness or severe sleepiness	1
16	P	5	None	1
17	P	4	nausea	1
18	P	1	nausea, depression	1
19	P	1	stomach pain, vomitting	1

```
Rate of the product is: 3
Product rate is satisfactory
PL/SQL procedure successfully completed.
```