



VIT[®]

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

DATABASE MANAGEMENT SYSTEM PROJECT

REVIEW 1

A PROJECT BY -:

RISHABH GAUBA-19BIT0313

UTKARSHGOYAL - 19BIT0402

SUKANT JHA - 19BIT0359

A project report on

ONLINE MATRIMONIAL SYSTEM

Submitted towards Jth component of the course

Database Management Systems – ITE-1003

handled by

PROF. BIMAL KUMAR RAY

submitted by

RISHABH GAUBA-19BIT0313

UTKARSHGOYAL - 19BIT0402

SUKANT JHA - 19BIT0359

PROBLEM DESCRIPTION

The main objective of online Matrimonial System is to provide Grooms and Brides with excellent matchmaking experience by exploring the opportunities and resources to meet true to potential partner. The drawback of existing system is that searching a good life partner in this world is the times consuming and cost effective. Online matrimonial system is used to overcome these drawbacks. This Online Matrimonial System is a useful for a person of any religion who wishes to find a suitable life partner for himself or herself. This website is very useful as it provides a fast searching of numerous suitable profiles.

INTRODUCTION

➤ ONLINE MATRIMONIAL SYSTEM

Online Matrimonial System provide platform to a lot of Bride/Groom for finding perfect match experience by exploring the opportunities and resources to meet true potential partner. Online matrimonial system is a project in which individual searching for their life partners can create an account and can search for their life partners according to their expectations. Keeping our objective in mind, we have created a renowned online matching services that will touch the souls of millions of people all over the globe.

➤ ADVANTAGES OF ONLINE MATRIMONIAL SYSTEM

Online matrimonial system is helps to search life partners according to one's expectations. There is a wide range of choice among which the bride or the groom can select the best one. Online matrimonial system is user friendly and safe. The bride or the groom can know about their partners by seeing the profile and further messaging. The system is convenient as well as economic. There is a lot of privacy which most of the people wish to have.

ANALYSIS

2.1 ENTITIES

- Profile
- User login
- Search profile
- Image gallery
- Success story
- Bank details

2.2 ATTRIBUTES

❖ PROFILE

- Profile_id
- Name
- Email_id
- Education
- Caste
- Country
- Height
- Occupation
- Weight
- Gothram
- Star

❖ USER LOGIN

- User_id
- User_password
- User_name

❖ SEARCH PROFILE

- Profile_id
- Search_pro_id
- Marital status
- Caste

❖ IMAGE GALLERY

- Profile_id
- Image_id

❖ SUCCESS STORY

- Success_profile_id
- Story
- Date_of_success

❖ BANK DETAILS

- Name_on_card
- Card_no
- Available_balance
- Card_type

2.3 RELATIONSHIP

Profile relationship with user login(1-1):-

There can be only be one profile for one user login person at once and only one person can have access to his/her own profile.

User login relationship with bank details(1-1):-

There can only be one user login accessing to his/her own bank details and bank details can be accessed by only one profile.

Profile relationship with Success story(1-1):-

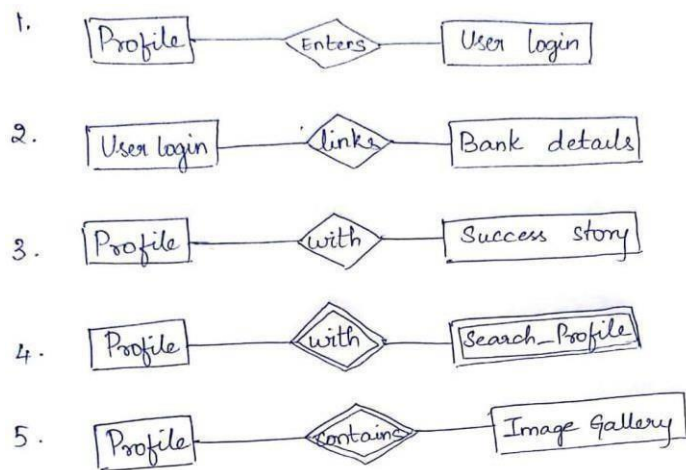
There can only be one kind of story for one profile and not the same story for the other profile. That's why it is one-to-one relationship.

Profile relationship with search profile (1-N):-

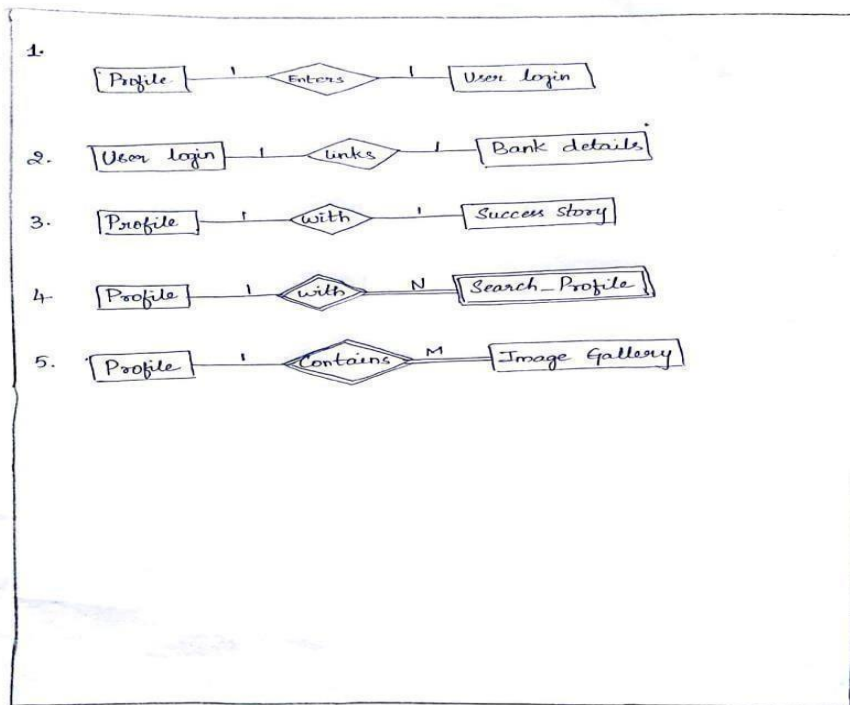
That unique profile can be searched by many other users on search profile. That's why it is 1-N relationship

Profile relationship with image gallery(1-M):-

Profile can have access to many images and many images can be inserted to one profile.



2.4 CARDINALITY



2.5 PARTICIPATION



Functional Requirements

Data Retrieval:-

1. The information of bride and groom sent to each other if there is a match successfully.
2. The customer might want to see the payment details and the profile information.
3. Details on different types of memberships with benefits for the customer to choose from.
4. Retrieval of data of those members who did not get a match yet and their subscription is also going to be over.
5. The customer might want the details of the employee working for the website who is finding a match for the bride/groom.
6. The customer wants the details on the number of views he/she got on the profile.
7. Details of the customer. For eg. If he/she has joined the site a very long time ago and has not been matched yet, so the site admin can give him/her some discount on the next subscription payment deadline.
8. The admin of the site can ask the customers to rate

the sites hospitality and functioning and reviews on what needs to be improved about the site.

Scenarios of modification of existing data:-

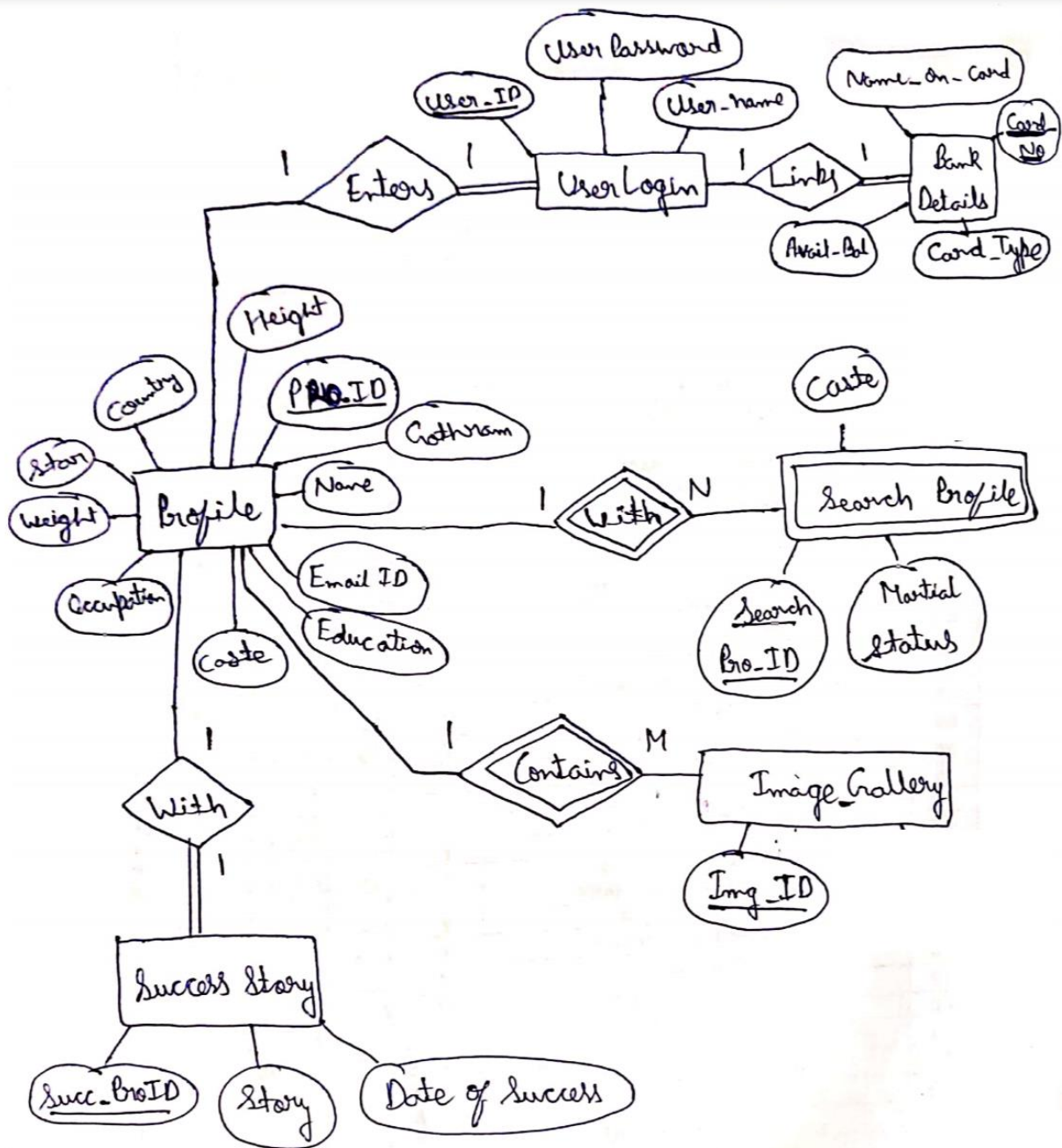
- 1.A customer might want to change or update his/her phone number and email address.
- 2.If the employee updates its profile, the customer should also know this as he/she is the one who is helping in finding a match.
- 3.A customer might want to update his/her hobbies, story or interest viewed by others.
- 4.A customer might want to set a new password or new username.
- 5.A customer can upgrade his/her membership.(for e.g. From Gold to Platinum with more benefits)

REMOVAL OF OLD DATA: -

1. A customer might want to delete his/her account.
2. The bride and grooms profile information and photo needs to be removed from the site once the match making is successful.
3. The system should remove the bank details of the customer once the matchmaking is successful to stop from further payment of the subscription every month.
4. Remove the employee details from the site if the employee has left his/her job.

DESIGN

ENTITY RELATIONSHIP DIAGRAM



REVIEW-2

A PROJECT BY -:

RISHABH GAUBA-19BIT0313

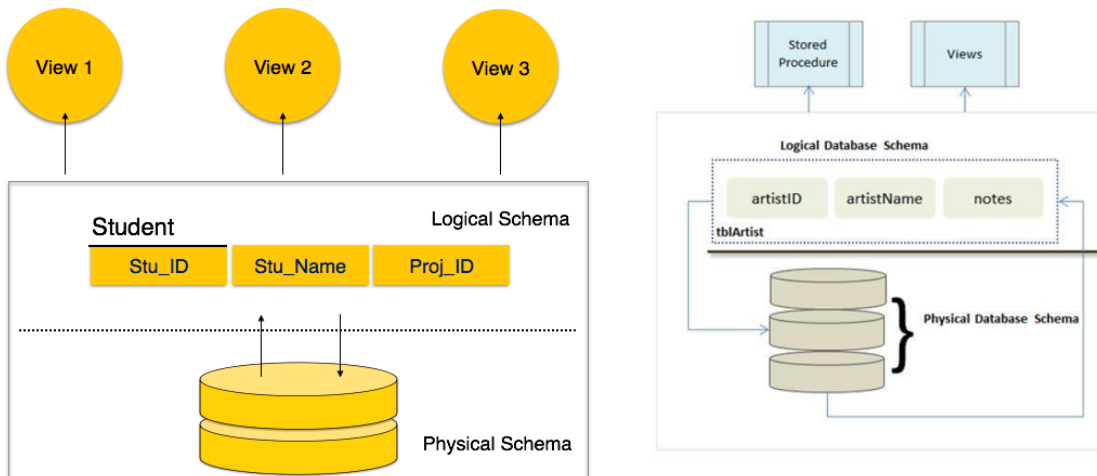
UTKARSHGOYAL - 19BIT0402

SUKANT JHA - 19BIT0359

What is a schema?

Database Schema

A database schema is the skeleton structure that represents the logical view of the entire database. It defines how the data is organized and how the relations among them are associated. It formulates all the constraints that are to be applied on the data. A database schema defines its entities and the relationship among them. It contains a descriptive detail of the database, which can be depicted by means of schema diagrams. It's the database designers who design the schema to help programmers understand the database and make it useful.



A database schema can be divided broadly into two categories –

Physical Database Schema – This schema pertains to the actual storage of data and its form of storage like files, indices, etc. It defines how the data will be stored in a secondary storage.

Logical Database Schema – This schema defines all the logical constraints that need to be applied on the data stored. It defines tables, views, and integrity constraints.

ER Model to Relational Model

ER Model, when conceptualized into diagrams, gives a good overview of entity-relationship, which is easier to understand. ER diagrams can be mapped to relational schema, that is, it is possible to create relational schema using ER diagram. We cannot import all the ER constraints into relational model, but an approximate schema can be generated.

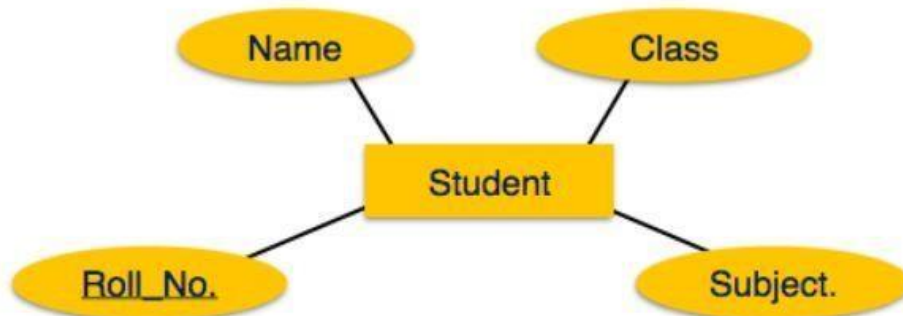
There are several processes and algorithms available to convert ER Diagrams into Relational Schema. Some of them are automated and some of them are manual. We may focus here on the mapping diagram contents to relational basics.

ER diagrams mainly comprise of –

- Entity and its attributes
- Relationship, which is association among entities.

Mapping Entity

An entity is a real-world object with some attributes.

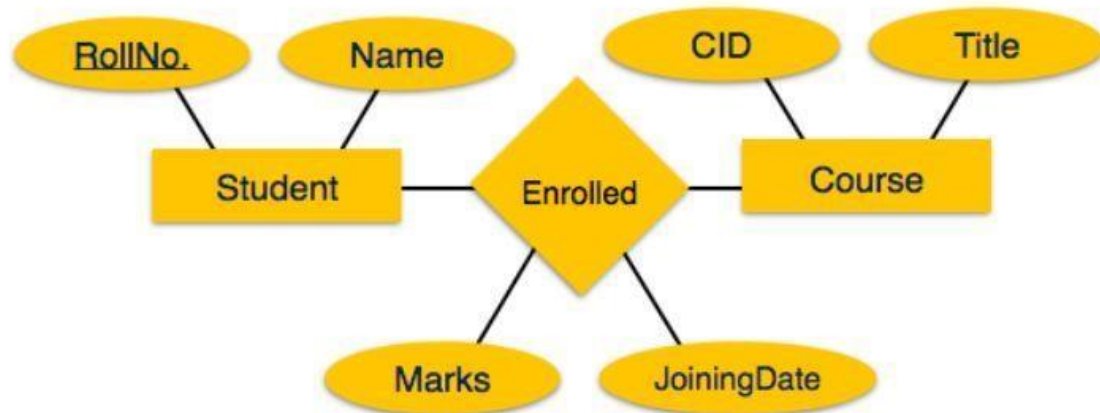


Mapping Process (Algorithm)

- ▣ Create table for each entity.
- ▣ Entity's attributes should become fields of tables with their respective data types.
- ▣ Declare primary key.

Mapping Relationship

A relationship is an association among entities.



Mapping Process

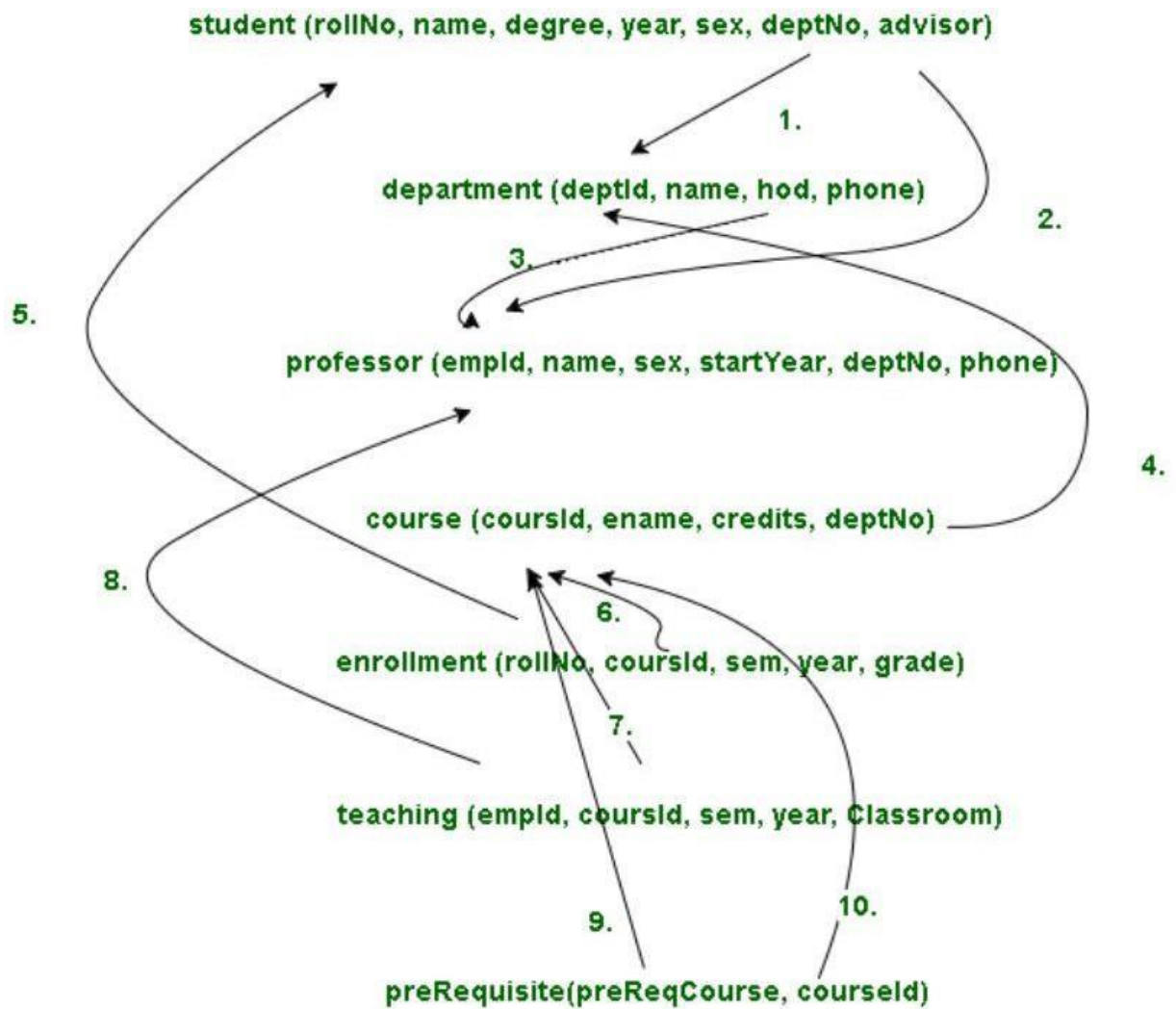
- Create table for a relationship.
- Add the primary keys of all participating Entities as fields of table with their respective data types.
- If relationship has any attribute, add each attribute as field of table.
- Declare a primary key composing all the primary keys of participating entities.
- Declare all foreign key constraints.

Relation Schema in DBMS

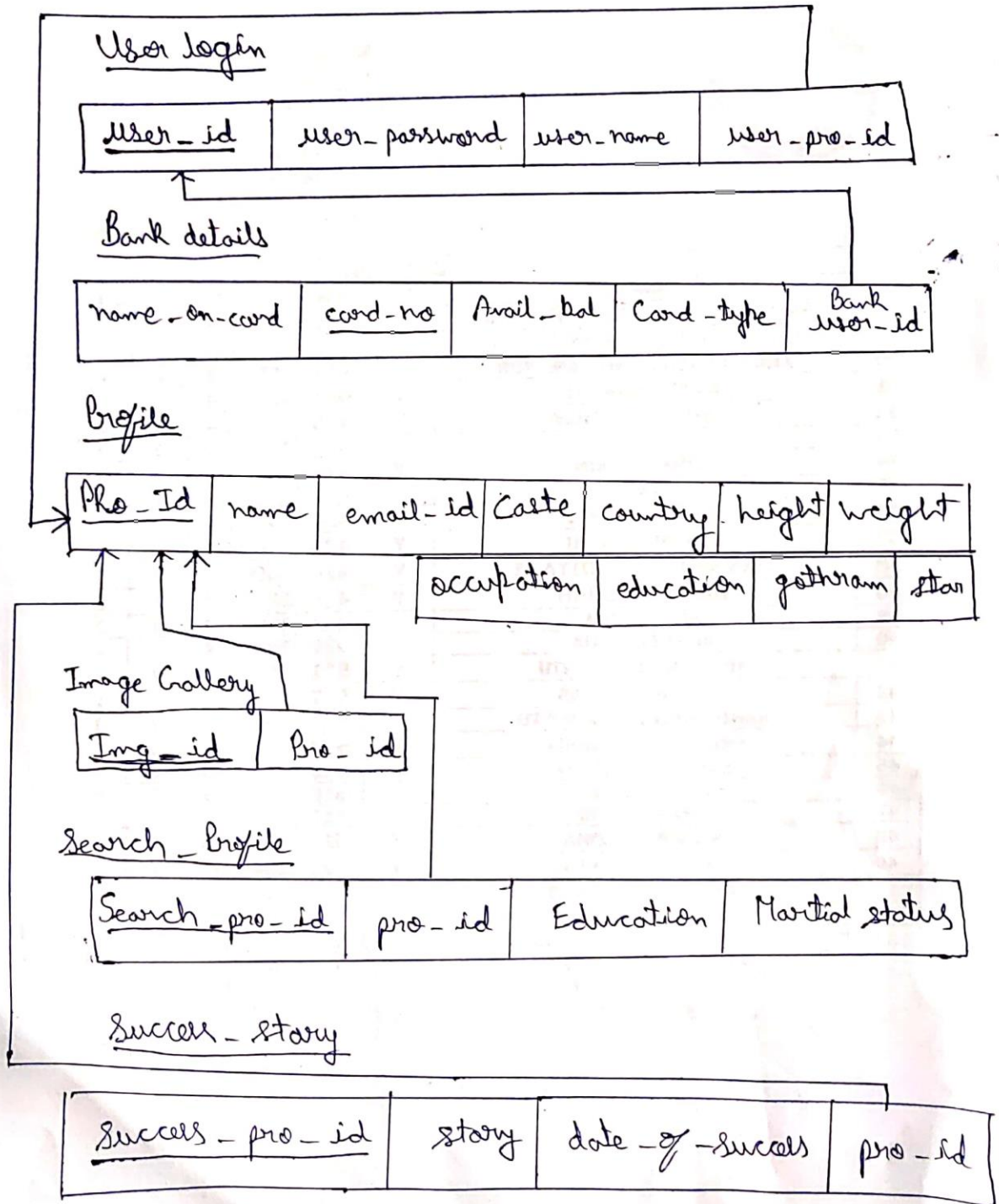
Relation schema defines the design and structure of the relation like it consists of the relation name, set of attributes/field names/column names. every attribute would have an associated domain.

There is a student named Geeks, she is pursuing B.Tech, in the 4th year, and belongs to IT department (department no. 1) and has roll number 1601347 She is proctored by Mrs. S Mohanty. If we want to represent this using databases we would have to create a student table with name, sex, degree, year, department, department number, roll number and proctor (adviser) as the attributes.

```
student (rollNo, name, degree, year, sex, deptNo, advisor)
```



MAPPING OF ER TO TABLE



IMPLEMENTATION

CREATE WITH CONSTRAINTS

```
create table user_login(  
2 user_id number(5) primary key,  
3 user_password varchar(10),  
4 user_name varchar(10));
```

```
SQL> create table user_login(  
2 user_id number(5) primary key,  
3 user_password varchar(10),  
4 user_name varchar(10));
```

Table created.

```
SQL> create table bank_details(  
2 name_on_card varchar(20),  
3 card_no number(10) primary key,  
4 ava_bal number(10),  
5 card_type varchar(20));
```

```
SQL> create table bank_details(  
2 name_on_card varchar(20),  
3 card_no number(10) primary key,  
4 ava_bal number(10),  
5 card_type varchar(20));
```

Table created.

Table created.

```
SQL> create table profile(  
2 pro_id number(5) primary key,  
3 name varchar(20),  
4 email_id varchar(50),  
5 caste varchar(10),  
6 country varchar(20),  
7 height varchar(30),  
8 weight varchar(30),  
9 education varchar(30),  
10 salary number(30),  
11 occupation varchar(10),  
12 gothram varchar(30),  
13 star varchar(30));
```

```
SQL> create table profile(  
  2  pro_id number(5) primary key,  
  3  name varchar(20),  
  4  email_id varchar(50),  
  5  caste varchar(10),  
  6  country varchar(20),  
  7  height varchar(30),  
  8  weight varchar(30),  
  9  education varchar(30),  
10  salary number(30),  
11  occupation varchar(10),  
12  gothram varchar(30),  
13  star varchar(30));
```

Table created.

```
create table image_gallery(  
  2  img_id number(5) primary key,  
  3  pro_id number(5),  
  4  foreign key(pro_id) references profile(pro_id));
```

```
SQL> create table image_gallery(  
  2  img_id number(5) primary key,  
  3  pro_id number(5),  
  4  foreign key(pro_id) references profile(pro_id));
```

Table created.

```
create table search_profile(  
  2  sear_pro_id number(5) primary key,  
  3  pro_id number(5),  
  4  education varchar(30),  
  5  martial_status varchar(20),  
  6  foreign key(pro_id) references profile(pro_id));
```

```
SQL> create table search_profile(  
  2  sear_pro_id number(5) primary key,  
  3  pro_id number(5),  
  4  education varchar(30),  
  5  martial_status varchar(20),  
  6  foreign key(pro_id) references profile(pro_id));
```

Table created.

```
create table success_story(  
  2  succ_pro_id number(5) primary key,  
  3  story varchar(30),  
  4  date_of_success date);
```

```
SQL> create table success_story(  
  2  succ_pro_id number(5) primary key,  
  3  story varchar(30),  
  4  date_of_success date);
```

Table created.

INSERT

Insert into user_login values(&user_id,&user_password,&user_name');

Insert into bank_details values('&name_on_card',&card_no,&ava_bal,&card_type');

Insert into profile

values(&pro_id,&name','&email_id','&caste','&country','&height','&weight','&education','&salary','&occupation','&gothram','&star');

Insert into image_gallery values(&img_id,&pro_id);

Insert into search_profile values(&sear_pro_id,&pro_id,&education','&marital_status');

Insert into success_story values(&succ_pro_id,&story','&date_of_success');

SQL Worksheet

```
1 create table user_login(  
2   user_id number(5) primary key,  
3   user_password varchar(10),  
4   user_name varchar(10));  
5 Insert into user_login values(12345,'29sne','sneha');  
6 Insert into user_login values(67890,'06rithu','rithu');  
7 select* from user_login;
```

Table created.

1 row(s) inserted.

1 row(s) inserted.

USER_ID	USER_PASSWORD	USER_NAME
12345	29sne	sneha
67890	06rithu	rithu

[Download CSV](#)

2 rows selected.

SQL Worksheet

```
1 create table bank_details(  
2   name_on_card varchar(20),  
3   card_no number(10) primary key,  
4   ava_bal number(10),  
5   card_type varchar(20));  
6 Insert into bank_details values('Sneha',295900,5000,'Mastercard');  
7 Insert into bank_details values('Rithu',166000,10000,'Visa');  
8 select*from bank_details;  
9
```

Table created.

1 row(s) inserted.

1 row(s) inserted.

NAME_ON_CARD	CARD_NO	AVA_BAL	CARD_TYPE
Sneha	295900	5000	Mastercard
Rithu	166000	10000	Visa

[Download CSV](#)

2 rows selected.

SQL Worksheet

 Clear

```
1 create table profile(  
2   pro_id number(5) primary key,  
3   name varchar(20),  
4   email_id varchar(50),  
5   caste varchar(10),  
6   country varchar(20),  
7   height varchar(30),  
8   weight varchar(30),  
9   education varchar(30),  
10  salary number(30),  
11  occupation varchar(10),  
12  gothram varchar(30),  
13  star varchar(30));  
14 insert into profile values(295,'Sneha','sneha43@gmail.com','Gandla','India',5.5,67,'B.Tech',100000,'Software','Irupudimahrshi','Gemini');  
15 insert into profile values(567,'Rithu','rithu78@gmail.com','Reddy','India',5.3,62,'B.Tech',150000,'Computer','Akshanthala','Scorpion');  
16 select*from profile;
```

Table created.

1 row(s) inserted.

1 row(s) inserted.

PRO_ID	NAME	EMAIL_ID	CASTE	COUNTRY	HEIGHT	WEIGHT	EDUCATION	SALARY	OCCUPATION	GOTHRAM	STAR
295	Sneha	sneha43@gmail.com	Gandla	India	5.5	67	B.Tech	100000	Software	Irupudimahrshi	Gemini
567	Rithu	rithu78@gmail.com	Reddy	India	5.3	62	B.Tech	150000	Computer	Akshanthala	Scorpion

[Download CSV](#)

2 rows selected.

SQL Worksheet

```
1 create table image_gallery(  
2   img_id number(5) primary key,  
3   pro_id number(5),  
4   foreign key(pro_id) references profile(pro_id));  
5 Insert into image_gallery values(9,295);  
6 Insert into image_gallery values(6,567);  
7 select*from image_gallery;  
8
```

Table created.

1 row(s) inserted.

1 row(s) inserted.

IMG_ID	PRO_ID
9	295
6	567

[Download CSV](#)

2 rows selected.

SQL Worksheet

```
1 create table search_profile(  
2 sear_pro_id number(5) primary key,  
3 pro_id number(5),  
4 education varchar(30),  
5 martial_status varchar(20),  
6 foreign key(pro_id) references profile(pro_id));  
7 Insert into search_profile values(169,295,'B.Tech Civil','Waiting');  
8 Insert into search_profile values(299,567,'B.Tech','Waiting');  
9 select*from search_profile;
```

Table created.

1 row(s) inserted.

1 row(s) inserted.

SEAR_PRO_ID	PRO_ID	EDUCATION	MARTIAL_STATUS
169	295	B.Tech Civil	Waiting
299	567	B.Tech	Waiting

[Download CSV](#)

2 rows selected.

SQL Worksheet

```
1 create table success_story(  
2 succ_pro_id number(5) primary key,  
3 story varchar(30),  
4 date_of_success date);  
5 Insert into success_story values(295,'Got married to ID172 lastmonth','11-OCT-18');  
6 Insert into success_story values(567,'Got married to ID789 lastmonth','20-OCT-18');  
7 select*from success_story;
```

ORA-00955: name is already used by an existing object

1 row(s) inserted.

1 row(s) inserted.

SUCC_PRO_ID	STORY	DATE_OF_SUCCESS
295	Got married to ID172 lastmonth	11-OCT-18
567	Got married to ID789 lastmonth	20-OCT-18

[Download CSV](#)

2 rows selected.

ALTER

Adding a suitable constraint for the email_id column of profile so that doesn't take any duplicates:

SQL Worksheet

```
1 alter table profile
2 modify email_id varchar(50) unique;
```

Table altered.

REVIEW-3

A PROJECT BY -:

RISHABH GAUBA-19BIT0313

UTKARSH GOYAL19BIT0402

SUKANT JHA - 19BIT0359

QUESTION-6

Write down the necessary SQL statements for implementation of functional requirements through SQL select, delete, alter and update statement.

UPDATE

1) Updating the search_profile to change the education details of a profile using their sear_pro_id:

SQL Worksheet

```
1 UPDATE search_profile
2 SET Education = 'B.com'
3 WHERE sear_pro_id = 189;
4 select*from search_profile;
```

1 row(s) updated.

SEAR_PRO_ID	PRO_ID	EDUCATION	MARTIAL_STATUS
169	295	B.Tech	Waiting
299	567	B.Tech	Waiting
189	678	B.com	Married

[Download CSV](#)

3 rows selected.

2) Update country in the profile using where clause.

SQL Worksheet

```
1 update profile
2 set country = 'Brazil'
3 where pro_id =567;
4 select*from profile;
5
```

1 row(s) updated.

PRO_ID	NAME	EMAIL_ID	CASTE	COUNTRY	HEIGHT	WEIGHT	EDUCATION	SALARY	OCCUPATION	GOTHRAM	STAR
295	Sneha	sneha43@gmail.com	Gandla	India	5.5	67	B.Tech	100000	Software	Irupudimahrshi	Gemini
567	Rithu	rithu78@gmail.com	Reddy	Brazil	5.3	62	B.Tech	150000	Computer	Akshanthala	Scorpion
389	Prity	prity57@gmail.com	Shrudas	Brazil	5.1	60	B.Tech	175000	Mechanical	Bharadwaja	Aquarius
589	Ravi	ravi89@gmail.com	Gandla	India	5.4	70	B.Sc	200000	Chemical	Jamadagni	Gemini

[Download CSV](#)
4 rows selected.

3) Update martial_status of the person after he or she gets married using their education details.

SQL Worksheet

```
1 update search_profile
2 set martial_status='Married'
3 where education='B.Sc';
4 select*from search_profile;
5
```

1 row(s) updated.

SEAR_PRO_ID	PRO_ID	EDUCATION	MARTIAL_STATUS
299	567	B.Tech	Waiting
980	389	B.Tech	Married
350	589	B.Sc	Married
169	295	B.Tech	Waiting

[Download CSV](#)
4 rows selected.

4)Update date of success using succ_pro_id foreign key.

SQL Worksheet

```
1 update success_story
2 set date_of_success='11-AUG-19'
3 where succ_pro_id=295;
4 select*from success_story;
```

1 row(s) updated.

SUCC_PRO_ID	STORY	DATE_OF_SUCCESS
295	Got married to ID172 lastmonth	11-AUG-19
567	Got married to ID789 lastmonth	20-OCT-18
389	Got married to ID890 lastmonth	23-FEB-20
589	Got married to ID908 lastmonth	12-AUG-19

[Download CSV](#)

4 rows selected.

DELETE

1)Deleting a row from search_profile using where clause and delete command:

SQL Worksheet

```
1 DELETE from search_profile
2 where pro_id=295;
```

```
1 row(s) deleted.
```

2)Delete row from profile using pro_id from the data.

SQL Worksheet

```
1 delete from success_story
2 where succ_pro_id=589;
3 select*from success_story;
```

1 row(s) deleted.

SUCC_PRO_ID	STORY	DATE_OF_SUCCESS
295	Got married to ID172 lastmonth	11-AUG-19
567	Got married to ID789 lastmonth	20-OCT-18
389	Got married to ID890 lastmonth	23-FEB-20

[Download CSV](#)

3 rows selected.

3)Delete all data from the database who's martial_status is 'waiting' and education from 'B.Sc' from search_profile table using 'where' and 'and' clause.

SQL Worksheet

```
1 DELETE FROM search_profile
2 WHERE education = 'B.Sc'
3 AND martial_status= 'Married';
4 select*from search_profile;
```

1 row(s) deleted.

SEAR_PRO_ID	PRO_ID	EDUCATION	MARTIAL_STATUS
299	567	B.Tech	Waiting
980	389	B.Tech	Married
169	295	B.Tech	Waiting

[Download CSV](#)

3 rows selected.

4)Delete all data from profile where star='Gemini'.

SQL Worksheet

```
1 Delete from profile
2 where star='Gemini';
3 select*from profile;
```

2 row(s) deleted.

PRO_ID	NAME	EMAIL_ID	CASTE	COUNTRY	HEIGHT	WEIGHT	EDUCATION	SALARY	OCCUPATION	GOTHRAM	STAR
567	Rithu	rithu78@gmail.com	Reddy	India	5.3	62	B.Tech	150000	Computer	Akshanthala	Scorpion
389	Prity	prity57@gmail.com	Shrudas	Brazil	5.1	60	B.Tech	175000	Mechanical	Bharadwaja	Aquarius

[Download CSV](#)

2 rows selected.

SELECT

1) Select profile details of all the users with their caste as “gandla”:

SQL Worksheet

```
1 select*from profile where caste='Gandla';
```

SQL Worksheet

```
1 select*from profile where caste='Gandla';
```

PRO_ID	NAME	EMAIL_ID	CASTE	COUNTRY	HEIGHT	WEIGHT	EDUCATION	SALARY	OCCUPATION	GOTHRAM	STAR
295	Sneha	sneha43@gmail.com	Gandla	India	5.5	67	B.Tech	100000	Software	Irupudimahrshi	Gemini
678	Ravi	ravi67@gmail.com	Gandla	America	5.9	70	B.com	200000	Mechanical	Kashyapa	Aquarius

[Download CSV](#)

2 rows selected.

2) Select statement used to display the number of sear_pro_id of each data to check whether the data is repeated or not.

SQL Worksheet

```
1 select count(sear_pro_id) from search_profile
2 GROUP BY sear_pro_id;
```

COUNT(SEAR_PRO_ID)
1
1
1
1

[Download CSV](#)

4 rows selected.

3) Select search_profile details of all the users with their education from “B.Sc”:-

SQL Worksheet

```
1 select*from search_profile where education='B.Sc';
```

SEAR_PRO_ID	PRO_ID	EDUCATION	MARTIAL_STATUS
350	589	B.Sc	Waiting

[Download CSV](#)

4) The following query counts the number of rows from two different tables (here we use profile and success story) using COUNT(*) command.

SQL Worksheet

```
1  SELECT(  
2      SELECT COUNT(*)  
3      FROM    profile  
4      ) AS Total_Customers,  
5      (SELECT COUNT(*)  
6      FROM    success_story  
7      ) AS No_Of_Stories  
8  FROM dual;  
9
```

TOTAL_CUSTOMERS	NO_OF_STORIES
6	4

[Download CSV](#)

Order By

1) Arrange table in ascending order of username.

SQL Worksheet

```
1 select * from user_login order by user_name;
```

USER_ID	USER_PASSWORD	USER_NAME
59087	05ravi	ravi
67890	06rithu	rithu
12345	29sne	sneha

[Download CSV](#)

3 rows selected.

2) Descending order

SQL Worksheet

```
1 select*from user_login order by user_name desc;
```

USER_ID	USER_PASSWORD	USER_NAME
12345	29sne	sneha
67890	06rithu	rithu
59087	05ravi	ravi

[Download CSV](#)

3 rows selected.

Group By

1) Select the no. of profiles whose marital status is “Waiting” and Education is ‘B.Tech’.

SQL Worksheet

```
1 select count(sear_pro_id) from search_profile
2 where education='B.Tech'
3 group by marital_status
4 having marital_status='Waiting';
```

COUNT(SEAR_PRO_ID)
2

[Download CSV](#)

SQL QUERIES USING JOIN/NESTING/SET OPERATIONS

SUBQUERY

Display name and caste from the profile whose image has its id as 6
[We find the relation between these two different tables and link them using their conditional sub-queries]

SQL Worksheet

```
1 select name,caste from profile
2 where pro_id=(select pro_id from image_gallery
3 where img_id=6);
```

NAME	CASTE
Rithu	Reddy

[Download CSV](#)

Full Outer join

1) Combination of profile table and search_profile table using the pro_id primary key.

SQL Worksheet

```
1 select*from profile full outer join search_profile on profile.pro_id=search_profile.pro_id;
```

PRO_ID	NAME	EMAIL_ID	CASTE	COUNTRY	HEIGHT	WEIGHT	EDUCATION	SALARY	OCCUPATION
295	Sneha	sneha43@gmail.com	Gandla	India	5.5	67	B.Tech	100000	Software
567	Rithu	rithu78@gmail.com	Reddy	India	5.3	62	B.Tech	150000	Computer
678	Ravi	ravi67@gmail.com	Gandla	America	5.9	70	B.com	200000	Mechanical

[Download CSV](#)

3 rows selected.

GOTHRAM	STAR	SEAR_PRO_ID	PRO_ID	EDUCATION	MARTIAL_STATUS
Irupudimahrshi	Gemini	169	295	B.Tech Civil	Waiting
Akshanthala	Scorpion	299	567	B.Tech	Waiting
Kashyapa	Aquarius	189	678	B.Tech	Married

Correlated Nested Query

1) Display the name of the person with gothram='Akshanthala' using correlated nested query.

SQL Worksheet

```
1 SELECT name FROM profile
2 WHERE pro_id=(
3 SELECT pro_id
4 FROM profile WHERE gothram='Akshanthala');
```

NAME
Rithu

[Download CSV](#)

2) Display name, email-id, country and gothram of the person who's martial_status='Waiting' using exists in the correlated query.

SQL Worksheet

```
1 SELECT name, country, gothram, email_id
2 FROM profile
3 WHERE exists (SELECT pro_id FROM search_profile
4 where search_profile.martial_status='Waiting'
5 and profile.pro_id=search_profile.pro_id);
```

NAME	COUNTRY	GOTHRAM	EMAIL_ID
Sneha	India	Irupudimahrshi	sneha43@gmail.com
Rithu	India	Akshanthala	rithu78@gmail.com

[Download CSV](#)

2 rows selected.

3) Martial_Status='Married'

SQL Worksheet

```
1 SELECT name, country, gothram, email_id
2 FROM profile
3 WHERE exists (SELECT pro_id FROM search_profile
4 where search_profile.martial_status='Married'
5 and profile.pro_id=search_profile.pro_id);
```

NAME	COUNTRY	GOTHRAM	EMAIL_ID
Ravi	America	Kashyapa	ravi67@gmail.com

[Download CSV](#)

Uncorrelated nested query

1) Display the name, country, education and gothram from profile through search profile table

SQL Worksheet

```
1 select name,country,education,gothram
2 from profile
3 where education in(select education from search_profile);
```

NAME	COUNTRY	EDUCATION	GOTHRAM
Sneha	India	B.Tech	Irupudimahrshi
Rithu	India	B.Tech	Akshanthala
Ravi	America	B.com	Kashyapa

Download CSV

3 rows selected.

Query involving any one of the SET Operators

1) Display the name and card_type of the person who has not payed through “mastercard” credit card.

SQL Worksheet

```
1 select name_on_card,card_type from bank_details
2 minus
3 select name_on_card,card_type from bank_details
4 where card_no in(select card_no from bank_details
5 where card_type='Mastercard');
```

NAME_ON_CARD	CARD_TYPE
Rithu	Visa

[Download CSV](#)

Join Query involving Order By clause

1) Joining of more than two tables: display the profile name,email_id,height,weight,salary,country,occupation,and goth ram according to the salary in descending order and martial_status='Waiting' to the number of times that profile searches for any other account, i.e the number of times the same details getting repeated shows that the user has searched for that many numbers of profiles.

SQL Worksheet

```
1 SELECT name,email_id,salary,country,gothram,occupation,height,weight
2 FROM profile
3 JOIN search_profile
4 ON profile.education = search_profile.education
5 where martial_status='Waiting'
6 ORDER BY profile.salary desc ;
```

NAME	EMAIL_ID	SALARY	COUNTRY	GOTHRAM	OCCUPATION	HEIGHT	WEIGHT
Rithu	rithu78@gmail.com	150000	India	Akshanthala	Computer	5.3	62
Rithu	rithu78@gmail.com	150000	India	Akshanthala	Computer	5.3	62
Sneha	sneha43@gmail.com	100000	India	Irupudimahrshi	Software	5.5	67
Sneha	sneha43@gmail.com	100000	India	Irupudimahrshi	Software	5.5	67

[Download CSV](#)

4 rows selected.

NULLIF FUNCTION

The NULLIF function compares two expressions. If they are equal, the function returns null. If they are not equal, the function returns the first expression. You cannot specify the literal NULL for first expression.

1) This code displays the length of name and gothram. If the length values are same the result will not show any value or null value, else name length value will be the output in the result column.

SQL Worksheet

```
1 SELECT LENGTH(name) "Name_length",  
2 LENGTH(gothram) "Gothram_length",  
3 NULLIF(LENGTH(name),LENGTH(gothram))  
4 result FROM profile;
```

Name_length	Gothram_length	RESULT
5	14	5
5	11	5
4	8	4

[Download CSV](#)

3 rows selected.

NVL FUNCTION

1) This SQL statement would return the user_name field if the user_password contained a null value. Otherwise, it would return the user_password.

SQL Worksheet

```
1 SELECT user_id,  
2 NVL(user_password,user_name)  
3 FROM user_login;  
4
```

USER_ID	NVL(USER_PASSWORD,USER_NAME)
12345	29sne
67890	06rithu
59087	05ravi

[Download CSV](#)

3 rows selected.

QUESTION-7

Define and implement two PL/SQL function involving cursor and two PL/SQL procedure involving cursor for the database under consideration

1) Display the name, pro_id ,country of all the people in the database present using PL/SQL cursor only.

SQL Worksheet

```
1 DECLARE
2     c_pro_id profile.pro_id%type;
3     c_name profile.name%type;
4     c_country profile.country%type;
5     CURSOR c_profile is
6         SELECT pro_id, name, country FROM profile;
7 BEGIN
8     OPEN c_profile;
9     LOOP
10        FETCH c_profile into c_pro_id, c_name, c_country;
11        EXIT WHEN c_profile%notfound;
12        dbms_output.put_line(c_pro_id || ' ' || c_name
13        || ' ' || c_country);
14    END LOOP;
15    CLOSE c_profile;
16 END;
17 /
```

Statement processed.

295 Sneha India

567 Rithu India

678 Ravi America

Function created with cursor

1)Return the maximum bank balance of a guy from a particular country return 0 if no data found.

SQL Worksheet

```
1 CREATE OR REPLACE FUNCTION
2 mb (c in profile.COUNTRY%type)
3 return bank_details.AVA_BAL%TYPE is
4 b bank_details.AVA_BAL%TYPE;
5 mx bank_details.AVA_BAL%TYPE:=0;
6 cursor cr is select *
7 from profile natural join user_login
8 where COUNTRY=c;
9 i cr%rowtype;
10 BEGIN
11 for i in cr loop
12 select AVA_BAL into b from bank_details
13 where bank_user_login=i.user_id;
14 if b>mx THEN
15 mx:=b;
16 end if;
17 end loop;
18 return mx;
19 end;
20 /
```

Function created.

```
21 declare
22 bal number(10);
23 country varchar(10);
24 begin
25 bal:=mb('India');
26 dbms_output.put_line(bal);
27 end;
28 /
```

Function created.

Statement processed.

95000

2) Return height of a person from profile id.

SQL Worksheet

```
1 CREATE OR REPLACE FUNCTION
2 he (p in profile.PRO_ID%TYPE)
3 return VARCHAR is
4 cursor cr is select * from profile
5 WHERE PRO_ID=p;
6 i cr%rowtype;
7 BEGIN
8 open cr;
9 fetch cr into i;
10 close cr;
11 return i.HEIGHT;
12 end;
13 /
14 select he(295) from dual;
```

Function created.

HE(295)
5.5

[Download CSV](#)

Procedure created with cursor

1)Return name and card number of the customer.

SQL Worksheet

```
1 CREATE OR REPLACE procedure
2 she is
3 cursor cr is select * from bank_details;
4 i cr%rowtype;
5 BEGIN
6 for i in cr loop
7 dbms_output.put_line
8 (
9 ' NAME: ' || i.NAME_ON_CARD || chr(10)
10 || 'Number: ' || i.CARD_NO || chr(10)
11 );
12 end loop;
13 end;
14 /
15 execute she;
16
```

Procedure created.

Statement processed.

NAME: Sneha

Number: 4567832

NAME: Rithu

Number: 5689456

NAME: Prity

Number: 9876543

NAME: Ravi

Number: 9436789

2) Display all the name and email id from profile in the database

SQL Worksheet

```
1 CREATE OR REPLACE procedure
2 pen is
3 cursor cr is select * from profile;
4 i cr%rowtype;
5 BEGIN
6 for i in cr loop
7 dbms_output.put_line
8 (
9 ' NAME: '||i.NAME||chr(10)
10 ||'Email ID: '||i.EMAIL_ID||chr(10)
11 );
12 end loop;
13 end;
14 /
15 execute pen;
16
```

Procedure created.

Statement processed.

NAME: Sneha
Email ID: sneha43@gmail.com

NAME: Rithu
Email ID: rithu78@gmail.com

NAME: Prity
Email ID: prity57@gmail.com

NAME: Ravi
Email ID: ravi89@gmail.com

QUESTION-8

Trigger

1) Display the error message if foreigners are more than 100 kg but if Indians are more than 100 kg than they are allowed in India.

SQL Worksheet

```
1 create or replace trigger coc
2 before insert on profile
3 for each row DECLARE
4 BEGIN
5 if :new.country<>'India' THEN
6 if :new.weight>100 THEN
7 raise_application_error
8 (
9     -20010,
10     'No foreigner can be obese'
11 );
12 end if;
13 end if;
14 end;
15 /
16 insert into profile(pro_id,country,weight) values(323,'sdf',134);
17
```

Trigger created.

ORA-20010: No foreigner can be obese ORA-06512: at "SQL_FFAVK00WKCUQRNQVJOQICRTL.N.COC", line 5
ORA-06512: at "SYS.DBMS_SQL", line 1721

SQL Worksheet

```
1 create or replace trigger coc
2 before insert on profile
3 for each row DECLARE
4 BEGIN
5 if :new.country<>'India' THEN
6 if :new.weight>100 THEN
7 raise_application_error
8 (
9     -20010,
10     'No foreigner can be obese'
11 );
12 end if;
13 end if;
14 end;
15 /
16 Insert into profile(pro_id,country,weight) values (435,'ssf',80);
```

Trigger created.

1 row(s) inserted.

SQL Worksheet

```
1 create or replace trigger coc
2 before insert on profile
3 for each row DECLARE
4 BEGIN
5 if :new.country<>'India' THEN
6 if :new.weight>100 THEN
7 raise_application_error
8 (
9     -20010,
10     'No foreigner can be obese'
11 );
12 end if;
13 end if;
14 end;
15 /
16 insert into profile(pro_id,country,weight) values(323,'India',134);
17
```

Trigger created.

1 row(s) inserted.

2) Display error message if visa card holders have more than 10000 in their account using trigger.

SQL Worksheet

```
1 create or replace trigger oo
2 before insert on bank_details
3 for each row DECLARE
4 BEGIN
5 if :new.CARD_TYPE='VISA' THEN
6 if :new.AVA_BAL>10000 THEN
7 raise_application_error
8 (
9 -20019,
10 'ACCOUNT having visa card
11 cannot have accout balance more than
12 10000'
13 );
14 end if;
15 end if;
16 end;
17 /
18 insert into bank_details(CARD_NO,CARD_TYPE,AVA_BAL) values(3434,'VISA',124342);
```

Trigger created.

ORA-20019: ACCOUNT having visa card
cannot have accout balance more than
10000 ORA-06512: at "SQL_FFAVKOOWKCUQRNQVJOQICRTL.N.BB", line 5
ORA-06512: at "SYS.DBMS_SQL", line 1721

SQL Worksheet

```
1 create or replace trigger oo
2 before insert on bank_details
3 for each row DECLARE
4 BEGIN
5 if :new.CARD_TYPE='VISA' THEN
6 if :new.AVA_BAL>10000 THEN
7 raise_application_error
8 (
9 -20019,
10 'ACCOUNT having visa card
11 cannot have accout balance more than
12 10000'
13 );
14 end if;
15 end if;
16 end;
17 /
18 insert into bank_details(CARD_NO,CARD_TYPE,AVA_BAL) values(344,'VISA',9800);
```

Trigger created.

1 row(s) inserted.

In this trigger the card type has been changed with more amount

of money added and still the database accepts it as the amount more than 10000 is not allowed only in the VISA card type but allowed in all other card types.

SQL Worksheet

```
1 create or replace trigger bb
2 before insert on bank_details
3 for each row DECLARE
4 BEGIN
5 if :new.CARD_TYPE='VISA' THEN
6 if :new.AVA_BAL>10000 THEN
7 raise_application_error
8 (
9     -20019,
10     'ACCOUNT having visa card
11     cannot have account balance more than
12     10000'
13 );
14 end if;
15 end if;
16 end;
17 /
18 insert into bank_details(card_no,card_type,ava_bal) values(4567321,'Mastercard',12000);
```

Trigger created.

1 row(s) inserted.

3) Display the error message if doctor's height is less than 170, else insert the row into the table.

SQL Worksheet

```
1 create or replace trigger eng
2 before insert on profile
3 for each row DECLARE
4 BEGIN
5 if :new.OCCUPATION='Doctor' THEN
6 if :new.HEIGHT<170 THEN
7 raise_application_error
8 (
9     -20019,
10     'Height of doctor must be
11     greater than 170'
12 );
13 end if;
14 end if;
15 end;
16 /
17 insert into profile(PRO_ID,HEIGHT,OCCUPATION) values(323,133,'Doctor');
```

Trigger created.

ORA-20019: Height of doctor must be
greater than 170 ORA-06512: at "SQL_FFAVKOONKCUQRNQVJOQICRTLN.ENG", line 5
ORA-06512: at "SYS.DBMS_SQL", line 1721

SQL Worksheet

```
1 create or replace trigger eng
2 before insert on profile
3 for each row DECLARE
4 BEGIN
5 if :new.OCCUPATION='Doctor' THEN
6 if :new.HEIGHT<170 THEN
7 raise_application_error
8 (
9     -20019,
10     'Height of doctor must be
11     greater than 170'
12 );
13 end if;
14 end if;
15 end;
16 /
17 insert into profile(pro_id,height,occupation) values(907,171,'Doctor');
```

Trigger created.

1 row(s) inserted.