DBMS PROJECT

Gym Management System Project

FACULTY- BIMAL KUMAR ROY

Submited by-

DEVESH GOYAL 17BIT0103

SHIV NARAYAN SINGH 17BIT0341

1).Choose a mini world for design and implementation of it's a database assigning an appropriate title for the database.

Ans)- Our proposed " Gym Management System" is for those who run a gym business. Before doing anything we did a decent research on major difficulties for gym owners. We examined carefully about how to make a huge registering system without failure as well as different functions for different kind of user depending on their privilege.

The Gym Management requires a system that will handle all the necessary and minute details easily and proper database security accordingly to the user. They requires software, which will store

data about members, employees, products, payroll, receipts of members & all transactions that occur in Gym.

2. Write down the data requirements and functional requirements for the database (in approximately 1500 words). The data requirements, apart from data to be stored in the database should also take into account the necessary integrity constraints that are reasonable for the database under consideration. The functional requirements should involve at least two different scenarios of removal of old data, at least two different scenarios for modification of existing data and four different scenarios of data retrieval.

Ans) **FUNCTIONAL REQUIREMENTS(MODULES):**

DASHBOARD

- Quick view of important Gym Modules
- Different report for management
- Calendar with important event and notices

GENERAL SETTINGS

- Admin can manage system Settings
- Set measurement units and header & fatter text as well
- Assign pay pal id for payment and set alert message

GYM MEMBER MODULE

- Record complete details of Gym subscriber
- Add vital measurement for each member
- Keep track of member's physical statistics

TRAINER MODULE

- Add different gym staff form admin

WORKOUT MODULE MEMBER

- Member can view the assigned workout

ACTIVITY MODULE

- Manage activities available in gym
- Assign activity to each membership plan

NUTRITION MODULE

- Prepare and assing nutrition plan for member

ATTENDANCE MODULE

- Manage staff attendance

GROUP MODULE

- Create and manage member groups

NOTICE MODULE

- Make important Announcement throught notice module

MESSAGE MODULE

- Admin can send message to all members

ACCOUNT MODULE

- Manage income and expenses
- Manage member fees

- Payment reports

CLASS MODULE

- Class Management
- Member list by class

FACILITY MANAGEMENT MODULE

- Record all the facilities available in gym

INVOICE MODULE

- Manage invoice from admin

ONLINE MEMBERSHIP PAYMENT MODULE

- Accept membership fees with paypal

REPORT MODULE

- Generate Report by membership plans
- Generate payment income report

CONSTRAINTS

Constraints are a very important feature in a relational model. In fact, the relational model supports the well-defined theory of constraints on attributes or tables. Constraints are useful because they allow a designer to specify the semantics of data in the database. *Constraints* are the rules that force DBMSs to check that data satisfies the semantics.

Integrity constraints provide a way of ensuring that changes made to the database by authorized users do not result in a loss of data consistency.

Domain Integrity

Domain restricts the values of attributes in the relation and is a constraint of the relational model.

Entity integrity

To ensure *entity integrity*, it is required that every table have a primary key. Neither the PK nor any part of it can contain null values. This is because null values for the primary key mean we cannot identify some rows.

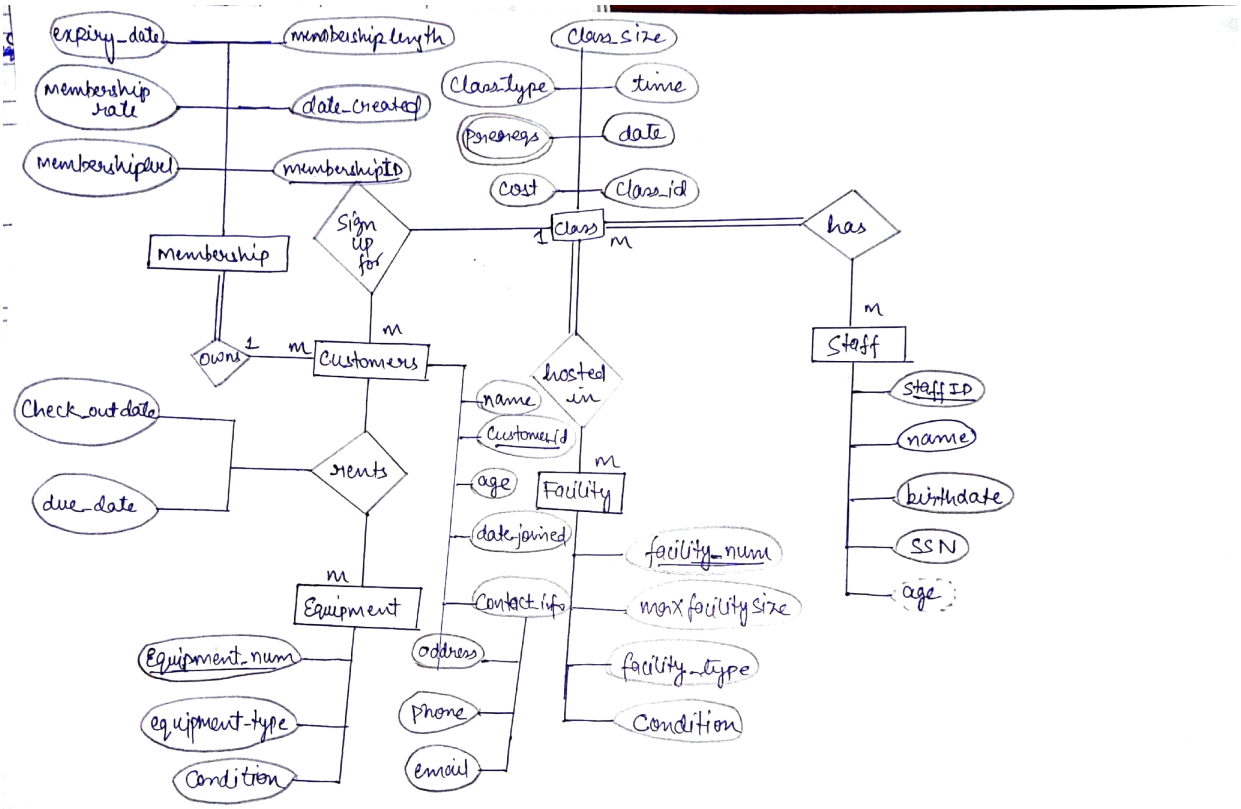
Referential integrity

Referential integrity requires that a foreign key must have a matching primarBy key or it must be null.

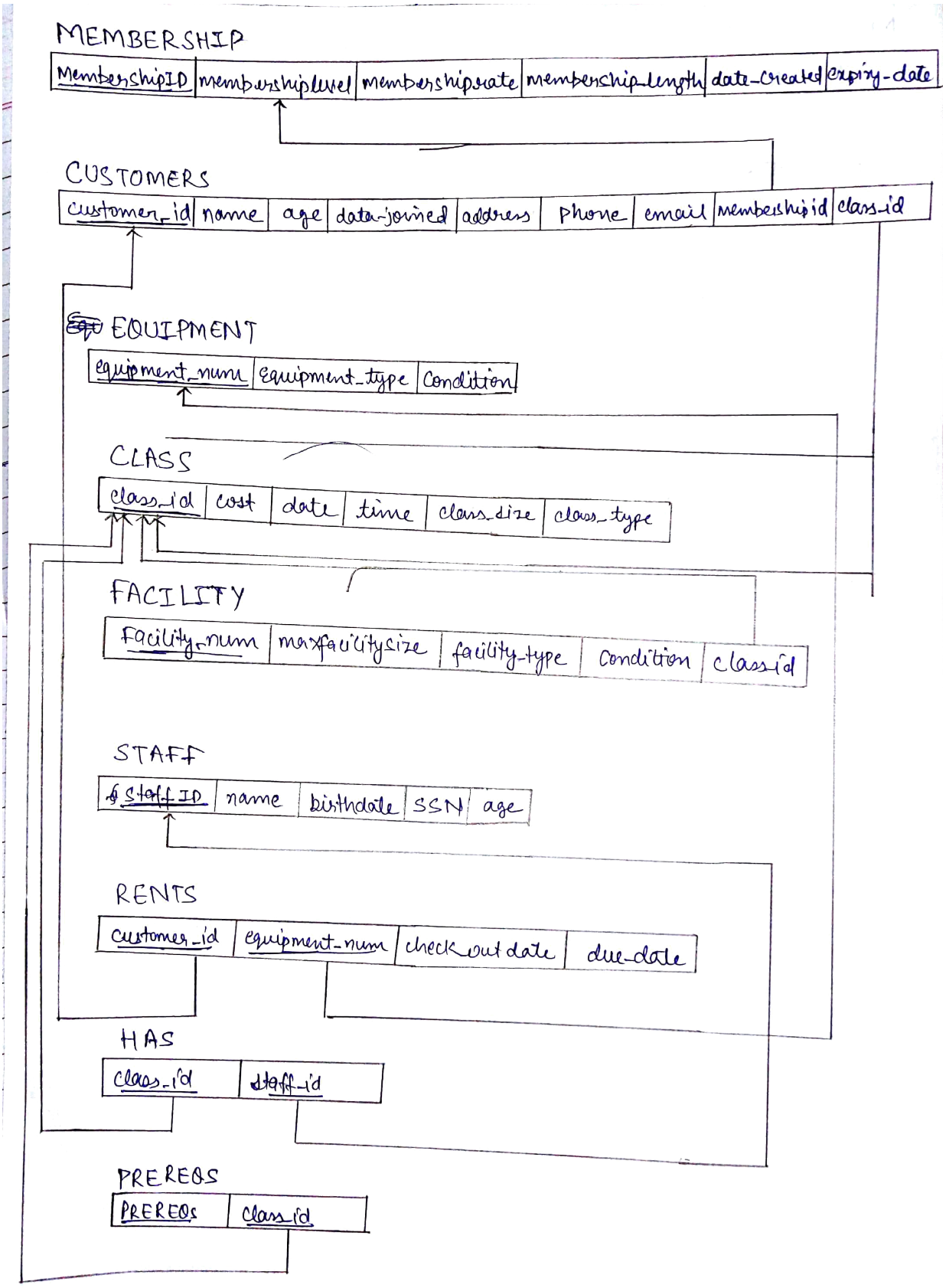
Description of Gym Management System Database:

- The details of Gym is store into Gym tables respective with all tables.
- Each entity (Member ,Trainer ,Branch ,Package ,Gym) contains primary key and unique keys.
- The entity Trainer, branch has binded with Gym , Package entities with foreign key.
- There is one-to-one and one-to-many relationships available between Branch , Payment , Member ,Gym.
- All the entities Gym ,Branch ,Trainer ,Member are normalized and reduce duplicacy of records.
- We have implemented indexing on each tables of Gym Management System tables for fast query execution.

E R DIAGRAM



RELATIONAL DATABASE


MEMBERSHIP
MembershipID
membershiplevel
membershiprate
membership_length
date_created
expiry_date
CUSTOMERS
customer_id
name
age
data_joined
address
phone
email
membership id
class_id
EQUIPMENT
equipment_num
Equipment_type
Condition
CLASS
class_id
cost
date
time
class_size
class_type
FACILITY
facility_num
maxfacilitysize
facility_type
Condition
class_id
STAFF
Staff ID
name
birthdate
SSN
age
RENTS
customer_id
equipment_num
check_out date
due_date
HAS
class_id
staff_id
PREREQS
PREREQs
class_id

5. Implement the relational database schema incorporating appropriate (based on data requirements) integrity constraints and enter necessary sample data into the tables. (15)

MEMBERSHIP

```
SQL>
SQL> desc membership
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 MEMBERSHIP_ID                             NOT NULL VARCHAR2(10)
 MEMBERSHIP_LEVEL                                   NUMBER(5)
 MEMBERSHIP_RATE                                    NUMBER(8)
 EXPIRY_DATE                                        DATE
 DATE_CREATED                                       DATE
 MEMBERSHIP_LENGTH                                  INTERVAL DAY(2) TO SECOND(6)

SQL> select * from membership;

MEMBERSHIP MEMBERSHIP_LEVEL MEMBERSHIP_RATE EXPIRY_DA DATE_CREA
---------- ---------------- --------------- --------- ---------
MEMBERSHIP_LENGTH
---------------------------------------------------------------------------
12345                    4            4500 04-MAY-18 04-AUG-18
+90 04:30:56.000000

23456                    2            2300 02-JAN-18 02-MAR-18
+60 03:28:24.000000

34567                    3            3500 22-OCT-18 22-JAN-18
+60 12:32:21.000000
```

CLASS

```
SQL> desc class
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 CLASS_ID                                  NOT NULL VARCHAR2(10)
 COST                                               NUMBER(8)
 CLASS_DATE                                         DATE
 TIME                                               TIMESTAMP(0)
 CLASS_SIZE                                         NUMBER(5)
 CLASS_TYPE                                         VARCHAR2(10)

SQL> select * from class;

CLASS_ID         COST CLASS_DAT
---------- ---------- ---------
TIME
---------------------------------------------------------------------------
CLASS_SIZE CLASS_TYPE
---------- ----------
classA           3200 12-MAY-18
12-MAY-18 12.56.32 PM
        10 special

classB           2000 11-MAY-18
11-MAY-18 11.22.23 AM
        10 regular

CLASS_ID         COST CLASS_DAT
---------- ---------- ---------
TIME
---------------------------------------------------------------------------
CLASS_SIZE CLASS_TYPE
---------- ----------

classC           1500 10-MAY-18
10-MAY-18 03.10.09 AM
        10 weekly
```

CUSTOMER

```
SQL> desc customer
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 CUSTOMER_ID                               NOT NULL VARCHAR2(10)
 NAME                                               VARCHAR2(20)
 AGE                                                NUMBER(3)
 DATE_OF_BIRTH                                      DATE
 ADDRESS                                            VARCHAR2(30)
 PHONE                                              NUMBER(10)
 EMAIL                                              VARCHAR2(10)
 MEMBERSHIP_ID                                      VARCHAR2(10)
 CLASS_ID                                           VARCHAR2(10)

SQL> select * from customer;

CUSTOMER_I NAME                          AGE DATE_OF_B
---------- -------------------- ---------- ---------
ADDRESS                        PHONE EMAIL      MEMBERSHIP CLASS_ID
------------------------------ ---------- ---------- ---------- ----------
customerA  Devesh                        19 22-APR-18
GandhiNagar                  9828283849 d@gmal.com 12345      classA

customerB  Shiv                          20 23-APR-18
Chennai                      9079842352 s@gmal.com 23456      classB

customerC  saketh                        22 23-APR-18
Kota                         9413583828 a@gmal.com 34567      classC
```

EQUIPMENT

```
SQL> desc  equipement
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 EQUIPEMENT_NUMBER                         NOT NULL NUMBER(2)
 EQUIPEMENT_TYPE                                    VARCHAR2(20)
 CONDITIONS                                         VARCHAR2(30)

SQL> select * from equipement;

EQUIPEMENT_NUMBER EQUIPEMENT_TYPE      CONDITIONS
----------------- -------------------- ------------------------------
                1 Exercise Bike        Registration
                2 Abdominal Exerciser  Age Restriction
                3 Cardio Machines      Payment
```

FACILITY

```
SQL> desc facility
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 FACILITY_NUMBER                           NOT NULL NUMBER(7)
 MAX_FACILITY_SIZE                                  INTERVAL DAY(2) TO SECOND(6)
 FACILITY_TYPE                                      VARCHAR2(10)
 CONDITIONS                                         VARCHAR2(10)
 CLASS_ID                                           VARCHAR2(10)

SQL> select * from facility
  2  ;

FACILITY_NUMBER
---------------
MAX_FACILITY_SIZE
---------------------------------------------------------------------------
FACILITY_T CONDITIONS CLASS_ID
---------- ---------- ----------
             1
+80 05:34:44.000000
CARDIO_THY Gym_Cloth  classA

             2
+45 02:20:25.000000
GRP_EXCSE  Regstrtion classB

FACILITY_NUMBER
---------------
MAX_FACILITY_SIZE
---------------------------------------------------------------------------
FACILITY_T CONDITIONS CLASS_ID
---------- ---------- ----------

             3
+90 04:56:23.000000
WOMENS_GYM Membership classC
```

STAFF

```
SQL> desc staff
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 STAFF_ID                                  NOT NULL VARCHAR2(10)
 NAME                                               VARCHAR2(20)
 DATE_OF_BIRTH                                      DATE
 SSN                                                NUMBER(6)
 AGE                                                NUMBER(3)
```

```
SQL> insert into staff values('staff01','sitanshu','17-FEB-2000','11','18');
insert into staff values('staff01','sitanshu','17-FEB-2000','11','18')
*
ERROR at line 1:
ORA-00001: unique constraint (SYSTEM.SYS_C007012) violated


SQL> insert into staff values('st01','sitanshu','17-FEB-1998','0011','20');

1 row created.

SQL> insert into staff values('st02','Dev','09-FEB-2000','0012','18');

1 row created.

SQL> insert into staff values('staff03','Gaurav','05-MAY-1999','13','19');

1 row created.

SQL> select *  from staff;

STAFF_ID   NAME                DATE_OF_B        SSN        AGE
---------- ------------------- --------- ---------- ----------
staff01    sitanshu            17-FEB-00          1         18
st01       sitanshu            17-FEB-98         11         20
st02       Dev                 09-FEB-00         12         18
staff03    Gaurav              05-MAY-99         13         19
```

RENT

```
SQL> desc rent
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 CUSTOMER_ID                                        VARCHAR2(10)
 EQUIPEMENT_NUMBER                                  NUMBER(2)
 CHECKOUT_DATE                                      DATE
 DUE_DATE                                           DATE
```

```
SQL> insert into rent values('customerA','01','09-APR-2018','19-APR-2018');

1 row created.

SQL> insert into rent values('customerB','02','10-MAR-2018','18-MAR-2018');

1 row created.

SQL> insert into rent values('customerC','03','13-MAY-2018','20-MAY-2018');

1 row created.

SQL> select * from rent;

CUSTOMER_I EQUIPEMENT_NUMBER CHECKOUT_ DUE_DATE
---------- ----------------- --------- ---------
customerA                  1 09-APR-18 19-APR-18
customerB                  2 10-MAR-18 18-MAR-18
customerC                  3 13-MAY-18 20-MAY-18
```

HAS

```
SQL> desc has
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 CLASS_ID                                           VARCHAR2(10)
 STAFF_ID                                           VARCHAR2(10)
```

```
SQL> insert into has values('classA','staff01');

1 row created.

SQL> insert into has values('classB','staff02');
insert into has values('classB','staff02')
*
ERROR at line 1:
ORA-02291: integrity constraint (SYSTEM.FK_STAFF) violated - parent key not
found


SQL> insert into has values('classA','staff02');
insert into has values('classA','staff02')
*
ERROR at line 1:
ORA-02291: integrity constraint (SYSTEM.FK_STAFF) violated - parent key not
found


SQL> insert into has values('classB','staff01');

1 row created.

SQL> insert into has values('classC','staff01');

1 row created.

SQL> select * from has;

CLASS_ID   STAFF_ID
---------- ----------
classA     staff01
classB     staff01
classC     staff01
```

PREREQS

```
SQL> create table prereqs(prereqs varchar(20) PRIMARY KEY,
  2    class_id,
  3    constraint tk_class foreign key (class_id) references class(class_id));

Table created.

SQL> insert into prereqs values('equipements','classA');

1 row created.

SQL> insert into prereqs values('trainer','classA)');
insert into prereqs values('trainer','classA)')
*
ERROR at line 1:
ORA-02291: integrity constraint (SYSTEM.TK_CLASS) violated - parent key not
found


SQL> insert into prereqs values('trainer','classB');

1 row created.

SQL> insert into prereqs values('manager','classC');

1 row created.

SQL> desc prereqs;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 PREREQS                                   NOT NULL VARCHAR2(20)
 CLASS_ID                                           VARCHAR2(10)

SQL> select * from prereqs;

PREREQS              CLASS_ID
-------------------- ----------
equipements          classA
trainer              classB
manager              classC
```

6. Write down the necessary SQL statements for implementation of functional requirements through SQL query, delete and update statement. (15)

MEMBERSHIP

```
SQL> update membership
  2  set m_rate=3700 where membership_id=34567;

1 row updated.

SQL> select * from membership;

MEMBERSHIP MEMBERSHIP_LEVEL     M_RATE EXPIRY_DA DATE_CREA
---------- ---------------- ---------- --------- ---------
MEMBERSHIP_LENGTH
---------------------------------------------------------------------------
12345                       4       4500 04-MAY-18 04-AUG-18
+90 04:30:56.000000

23456                       2       2300 02-JAN-18 02-MAR-18
+60 03:28:24.000000

34567                       3       3700 22-OCT-18 22-JAN-18
+60 12:32:21.000000
```

EQUIPMENT

```
SQL> update equipement
  2  set equipement_type='Stretching machines' where equipement_number=2;

1 row updated.

SQL> select * from equipement;

EQUIPEMENT_NUMBER EQUIPEMENT_TYPE      CONDITIONS
----------------- -------------------- ------------------------------
                1 Exercise Bike        Registration
                2 Stretching machines  Age Restriction
                3 Cardio Machines      Payment
```

CLASS

```
SQL> update class
  2  set class_type='AC regular' where cost=3200;

1 row updated.

SQL> select * from class;

CLASS_ID         COST CLASS_DAT
---------- ---------- ---------
TIME
---------------------------------------------------------------------------
CLASS_SIZE CLASS_TYPE
---------- ----------
classA           3200 12-MAY-18
12-MAY-18 12.56.32 PM
        10 AC regular

classB           2000 23-AUG-18
11-MAY-18 11.22.23 AM
        10 regular

CLASS_ID         COST CLASS_DAT
---------- ---------- ---------
TIME
---------------------------------------------------------------------------
CLASS_SIZE CLASS_TYPE
---------- ----------

classC           1500 10-MAY-18
10-MAY-18 03.10.09 AM
        10 weekly
```

FACILTY

```
SQL> update facility
  2  set facility_type='Mscl_bld' where facility_number=2;

1 row updated.

SQL> update facility
  2  set facility_type='Hrt_exc' where class_id='classA';

1 row updated.

SQL> select * from facility;

FACILITY_NUMBER
---------------
MAX_FACILITY_SIZE
--------------------------------------------------------------------------------
FACILITY_T CONDITIONS CLASS_ID
---------- ---------- ----------
             1
+80 05:34:44.000000
Hrt_exc    Gym_Cloth  classA

             2
+45 02:20:25.000000
Mscl_bld   Regstrtion classB

FACILITY_NUMBER
---------------
MAX_FACILITY_SIZE
--------------------------------------------------------------------------------
FACILITY_T CONDITIONS CLASS_ID
---------- ---------- ----------

             3
+90 04:56:23.000000
WOMENS_GYM Membership classC
```

```
SQL> delete from facility
  2  where class_id='classC';

1 row deleted.

SQL> select * from facility;

FACILITY_NUMBER
---------------
MAX_FACILITY_SIZE
--------------------------------------------------------------------------------
FACILITY_T CONDITIONS CLASS_ID
---------- ---------- ----------
             1
+80 05:34:44.000000
Hrt_exc    Gym_Cloth  classA

             2
+45 02:20:25.000000
Mscl_bld   Regstrtion classB

FACILITY_NUMBER
---------------
MAX_FACILITY_SIZE
--------------------------------------------------------------------------------
FACILITY_T CONDITIONS CLASS_ID
---------- ---------- ----------
```

STAFF

```
SQL> update staff
  2  set name='Shiv' where SSN=1;

1 row updated.

SQL> update staff
  2  set name='Devesh' where staff_id='st02';

1 row updated.

SQL> select * from staff;

STAFF_ID   NAME                DATE_OF_B        SSN        AGE
---------- ------------------- --------- ---------- ----------
staff01    Shiv                17-FEB-00          1         18
st01       sitanshu            17-FEB-98         11         20
st02       Devesh              09-FEB-00         12         18
staff03    Gaurav              05-MAY-99         13         19
```

RENT

```
SQL> select * from rent;

CUSTOMER_I EQUIPEMENT_NUMBER CHECKOUT_ DUE_DATE
---------- ----------------- --------- ---------
customerA                  1 09-APR-18 19-APR-18
customerB                  2 12-OCT-18 22-OCT-18

SQL> update rent
  2  set checkout_date='10-APR-18' where customer_id='customerA';

1 row updated.

SQL> update rent
  2  set due_date='20-OCT-18' where equipement_number=2;

1 row updated.

SQL> select * from rent;

CUSTOMER_I EQUIPEMENT_NUMBER CHECKOUT_ DUE_DATE
---------- ----------------- --------- ---------
customerA                  1 10-APR-18 19-APR-18
customerB                  2 12-OCT-18 20-OCT-18
```

HAS

```
SQL> select * from has;

CLASS_ID   STAFF_ID
---------- ----------
classA     staff01
classB     st01

SQL> update has
  2  set class_id='classC' where staff_id='st01';

1 row updated.

SQL> delete from has where class_id='classA';

1 row deleted.
```

PREREQS

```
SQL> select * from prereqs;

PREREQS              CLASS_ID
-------------------- ----------
trainer              classB
manager              classC

SQL> update prereqs
  2  set prereqs='recep' where class_id='classC';

1 row updated.

SQL> delete from prereqs where class_id='classB';

1 row deleted.

SQL>
```

QUERIES

SELECT WHERE CLAUSE

```
Connected.
SQL> select membership_level,m_rate from membership where membership_id='12345';

MEMBERSHIP_LEVEL     M_RATE
---------------- ----------
               4       4500

SQL> select cost,class_size from class where class_id='classA';

      COST CLASS_SIZE
---------- ----------
      3200         10

SQL> select name,age,email from customer where customer_id='customerA';

NAME                        AGE EMAIL
-------------------- ---------- ----------
Devesh                       19 d@gmal.com

SQL> select equipement_type from equipement where equipement_number=1;

EQUIPEMENT_TYPE
--------------------
Exercise Bike

SQL> select facility_type, max_facility_size from facility where facility_number=1;

FACILITY_T
----------
MAX_FACILITY_SIZE
---------------------------------------------------------------------------
Hrt_exc
+80 05:34:44.000000


SQL> select name,date_of_birth from staff where staff_id='st01';

NAME                 DATE_OF_B
-------------------- ---------
sitanshu             17-FEB-98

SQL>
```

COMPARISION OPERATOR

```
SQL> select * from equipement where equipement_number<3;

EQUIPEMENT_NUMBER EQUIPEMENT_TYPE      CONDITIONS
----------------- -------------------- ----------------------------
                1 Exercise Bike        Registration
                2 Stretching machines  Age Restriction

SQL> select * from facility where facility_number between 1 and 3;

FACILITY_NUMBER
---------------
MAX_FACILITY_SIZE
--------------------------------------------------------------------------------
FACILITY_T CONDITIONS CLASS_ID
---------- ---------- ----------
             1
+80 05:34:44.000000
Hrt_exc    Gym_Cloth  classA

             2
+45 02:20:25.000000
Mscl_bld   Regstrtion classB

FACILITY_NUMBER
---------------
MAX_FACILITY_SIZE
--------------------------------------------------------------------------------
FACILITY_T CONDITIONS CLASS_ID
---------- ---------- ----------


SQL> select * from staff where staff_id is not NULL;

STAFF_ID   NAME                 DATE_OF_B        SSN        AGE
---------- -------------------- --------- ---------- ----------
staff01    Shiv                 17-FEB-00          1         18
st01       sitanshu             17-FEB-98         11         20
st02       Devesh               09-FEB-00         12         18
staff03    Gaurav               05-MAY-99         13         19

SQL>
```

AGGREGATE FUNCTION

```
SQL> select avg(m_rate) from membership;

AVG(M_RATE)
-----------
       3500

SQL>
SQL> select min(m_rate) from membership;

MIN(M_RATE)
-----------
       2300

SQL> select count(class_id) from class;

COUNT(CLASS_ID)
---------------
              5

SQL> select stddev(cost) from class;

STDDEV(COST)
------------
  1619.56784

SQL>
```

NUMERIC FUNCTIONS

```
SQL> select abs(facility_number) from facility;

ABS(FACILITY_NUMBER)
--------------------
                   1
                   2

SQL> select sqrt(cost) from class;

SQRT(COST)
----------
56.5685425
        50
38.7298335
67.0820393
74.8331477

SQL> select floor(equipement_number) from equipement;

FLOOR(EQUIPEMENT_NUMBER)
------------------------
                       1
                       2
                       3

SQL> select ceil(facility_number) from facility;

CEIL(FACILITY_NUMBER)
---------------------
                    1
                    2

SQL>
```

STRING FUNCTIONS

```
SQL> select upper(name) from staff;

UPPER(NAME)
--------------------
SHIV
SITANSHU
DEVESH
GAURAV

SQL> select lower(name) from customer;

LOWER(NAME)
--------------------
devesh
shiv
saketh

SQL> select initcap(name) from customer;

INITCAP(NAME)
--------------------
Devesh
Shiv
Saketh

SQL> select reverse(name) from staff;

REVERSE(NAME)
--------------------
vihS
uhsnatis
hseveD
varuaG

SQL> select length(name) from customers;

no rows selected

SQL> select length(name) from customer;

LENGTH(NAME)
------------
           6
           4
           6

SQL>
```

SET OPERATIONS

```
SQL> select membership_id from membership union select class_id from class;

MEMBERSHIP
----------
12345
23456
34567
classA
classB
classC
classD
classE

8 rows selected.

SQL> select customer_id  from customer intersect select facility_number from facility;
select customer_id  from customer intersect select facility_number from facility
       *
ERROR at line 1:
ORA-01790: expression must have same datatype as corresponding expression


SQL> select membership_id from membership intersect select class_id from class;

no rows selected

SQL> select facility_number from facility minus select equipement_number from equipement;

no rows selected
```

GROUP BY OR ORDER BY

```
SQL> select * from facility order by class_id;

FACILITY_NUMBER
---------------
MAX_FACILITY_SIZE
--------------------------------------------------------------------------------
FACILITY_T CONDITIONS CLASS_ID
---------- ---------- ----------
             1
+80 05:34:44.000000
Hrt_exc    Gym_Cloth  classA

             2
+45 02:20:25.000000
Mscl_bld   Regstrtion classB

FACILITY_NUMBER
---------------
MAX_FACILITY_SIZE
--------------------------------------------------------------------------------
FACILITY_T CONDITIONS CLASS_ID
---------- ---------- ----------


SQL> select * from customer order by class_id;

CUSTOMER_I NAME                      AGE DATE_OF_B
---------- -------------------- ---------- ---------
ADDRESS                         PHONE EMAIL      MEMBERSHIP CLASS_ID
------------------------------ ---------- ---------- ---------- ----------
customerA  Devesh                       19 22-APR-18
GandhiNagar                   9828283849 d@gmal.com 12345      classA

customerB  Shiv                         20 23-APR-18
Chennai                       9079842352 s@gmal.com 23456      classB

customerC  saketh                       22 23-APR-18
Kota                          9413583828 a@gmal.com 34567      classC


SQL>
```

SUB QUERIES

```
SQL> select equipement_type from equipement  where equipement_number>(select equipement_number from rent where customer_id='12345');

no rows selected

SQL> select equipement_type from equipement where equipement_number<(select equipement_number from rent where customer_id='12345');

no rows selected
```

JOIN OPERATORS

```
SQL> select customer.customer_id,customer.name,customer.age,class.class_id,class.cost from customer
  2  inner join class on customer.class_id = class.class_id;

CUSTOMER_I NAME                      AGE CLASS_ID        COST
---------- -------------------- ---------- ---------- ----------
customerA  Devesh                       19 classA           3200
customerB  Shiv                         20 classB           2500
customerC  saketh                       22 classC           1500

SQL> select facility.facility_number,facility.facility_type,class.cost,class.class_date,class.class_size from facility
  2  right join class on facility.class_id = class.class_id
  3  order by facility.facility_number;

FACILITY_NUMBER FACILITY_T       COST CLASS_DAT CLASS_SIZE
--------------- ---------- ---------- --------- ----------
              1 Hrt_exc          3200 12-MAY-18         10
              2 Mscl_bld         2500 23-AUG-18         10
                                 5600 06-AUG-18         30
                                 4500 05-AUG-18         23
                                 1500 10-MAY-18         10

SQL>
```

CREATE VIEWS

```
                                 1500 10-MAY-18         10

SQL> create or replace view view1 as select max_facility_size,facility_type,class_id from facility where facility_number>=1;

View created.

SQL> create or replace view view1 as select max_facility_size,facility_type,class_id from facility where facility_number>=2;

View created.
```

7. Define and implement one PL/SQL function and one PL/SQL procedure appropriate for the database under consideration. (10)

PL/SQL FUNCTION

Set serveroutput on;
Declare
Membership1 number(5);
Membership2 number(5);
Result number(5);
Function pricefind(x in number,y in number)
Return number
Is
Z number;
Begin
If x<y then
Z:=x;
Else
Z:=y;
End if;
Return z;
End;
Begin
Select m_rate into membership1 from membership where membership_id=’12345’;

Select m_rate into membership2 from membership where membership_id=’23456’;
Result:=pricefind(membership1,membership2);
Dbms_output.put_line(‘the lowest m_rate is’||result);
End;
/

```
SQL> set serveroutput on;
SQL>  declare
  2   membership1 number(5);
  3   membership2 number(5);
  4   result number(5);
  5   function pricefind(x in number,y in number)
  6   return number
  7   is
  8   z number;
  9   begin
 10   if x<y then
 11   z:=x;
 12   else
 13   z:=y;
 14   end if;
 15   return z;
 16   end;
 17   begin
 18   select m_rate into membership1 from membership where membership_id='12345';
 19   select m_rate into membership2 from membership where membership_id='23456';
 20   result:=pricefind(membership1,membership2);
 21   dbms_output.put_line('the lowest m_rate is'||result);
 22   end;
 23  /
the lowest m_rate is2300

PL/SQL procedure successfully completed.
```

PL/SQL procedure

Set serveroutput on;
Declare
Membership1 number(5);
Membership2 number(5);
Result number(5);
Procedure pricefind(x in number,y in number,z out number) is
Begin
If x>y then
Z:=x;
Else
Z:=y;
End if;
End;
Begin
Select m_rate into membership1 from membership where membership_id=’12345’;
Select m_rate into membership2 from membership where membership_id=’34567’;
pricefind(membership1,membership2,result);

Dbms_output.put_line('the m_rate is'||result);
End;
/

```
SQL> set serveroutput on;
SQL>  declare
  2   membership1 number(5);
  3   membership2 number(5);
  4   result number(5);
  5   procedure pricefind(x in number,y in number,z out number) is
  6   begin
  7   if x>y then
  8   z:=x;
  9   else
 10   z:=y;
 11   end if;
 12   end;
 13   begin
 14   select m_rate into membership1 from membership where membership_id='12345';
 15   select m_rate into membership2 from membership where membership_id='34567';
 16   pricefind(membership1,membership2,result);
 17   dbms_output.put_line('the m_rate is'||result);
 18   end;
 19  /
the m_rate is4500

PL/SQL procedure successfully completed.
```

8. Define two business rules appropriate for the database under consideration and implement the rules using trigger. (15)

```
SQL> create or replace trigger m_rate_changes
  2   before insert or update on membership
  3   for each row
  4   when(NEW.membership_id > 11111)
  5   declare
  6   m_rate_diff number;
  7   begin
  8   m_rate_diff := :NEW.m_rate - :OLD.m_rate;
  9   dbms_output.put_line('OLD rate:'|| :OLD.m_rate);
 10   dbms_output.put_line('NEW rate:'|| :NEW.m_rate);
 11   dbms_output.put_line('Rate difference:'||m_rate_diff);
 12   end;
 13  /

Trigger created.

SQL>
```

```
SQL>  create or replace trigger c_rate_changes
  2     before insert on class
  3     for each row
  4     declare
  5     rowcount number;
  6     begin
  7     select count(*) into rowcount from class where cost = :NEW.cost;
  8     if rowcount <> 0 then
  9     raise_application_error(-78165,'error');
 10    end if;
 11  end;
 12  /

Trigger created.

SQL>
SQL>
SQL> insert into class values('classA',3200,'12-MAY-18','12-MAY-18 12:56:32',10,'special');
insert into class values('classA',3200,'12-MAY-18','12-MAY-18 12:56:32',10,'special')
           *
ERROR at line 1:
ORA-21000: error number argument to raise_application_error of -78165 is out of
range
ORA-06512: at "SYSTEM.C_RATE_CHANGES", line 6
ORA-04088: error during execution of trigger 'SYSTEM.C_RATE_CHANGES'
```