1.INTRODUCTION(problem description)

In educational institutions, online sessions usually follow a time table to attend classes. Recorded classes are exceptional, generally it became a reason for procrastination, off the topic. What if the time table concept does not exist in our world? What will be the outcome? There is no exact time to enter the college or attend sessions for students as well as teachers or staff. This causes a huge confusion right. Example you can't sit in college at 3:00 am to learn probability and statistics.

Then assume that the time table exists in our life, it actually exists, but there will be huge problems if we cannot manage them. For suppose you made a time table for 5 th class students but the staff of that class assigned with any random class time table. Although time tables are created, they are not able to manage efficiently. Also it will become difficult to assign a substitute teacher because of the absence of the Time table Management System.

One of the solutions to this huge headache is the Time Management System which can be created and maintained by Database Management System. By this method we are able to assign a time table to each person . Also we can check the vacancy of class and assign substitute teachers to that class. We are presenting a TTMS(Time table Management System) ,here we considered the scenario, online education .

2.DESCRIPTION ON DATABASE

There are many ways to describe this Database,let us consider one of the many strings to this design. This string contains the following points mentioned below:

- The details of Timetable are stored into the Timetable tables respective with all tables.
- Each entity (Teacher, Class, Subject, Attendance, Timetable) contains primary key and unique keys.
- The entity Class, Subject has binded with Timetable, Attendance entities with foreign key
- There are one-to-one and one-to-many tableships available between Subject, Student,Teacher, Timetable.

3.STEPS TO BUILD DATABASE

There are some general step steps to follow in the path of creating a database.Some of the general steps are mentioned below:

1. **Determine the requirements for the database:** Before building a database, it is important to consider the requirements and needs of the problem statement.It helps us to know what information needs to be stored and what it is used for later. Identify the entities and constraints that exist between them.
2. **Design the database schema:** Once the requirements are specified, the next step is to design the database schema. The design includes the proper overview of the relationships and relations. The ER[Entity relationship] model is used , it gives the overview of the entities and their attributes.

The ER model is then converted to conceptual database design which actually shows the total amount of tables that need to be created. This situation exists due to the rise of multiple attributes and many-to-many relationship between entities.

3. **Create a database:**Use of DBMS such as Oracle, MySQL, or SQL server to create the physical database.This step involves creating the tables and fields defined in the schema.In this project Oracle 10g and Live SQL oracle server is used in the process of creating the database.The general DDL[Data Definition Language] commands used in this project are mentioned below:

- **Syntax to create table:**

```
CREATE TABLE table_name (
   column1 data type constraint,
   column2 data type constraint,
   ...
   column_n data type constraint
);
```

- **Syntax to alter table:**

```
ALTER TABLE table_name
{ ADD | MODIFY | DROP } column_name data type constraint;
```

4. **Populate the database with insertion initial data:**Once the schema is created the data needs to be inserted in the database.This can be done manually by using insert statements or we can import a file or using a script we a=can extract data. This project manually inserted data by using insert statements.To DML[Data Manipulation Language] is used to insert data, the general syntax is given below:

- **Syntax to alter table:**

```
INSERT INTO table_name (column1, column2, column3, ...) VALUES (value1, value2, value3, ...);
```

5. **Test the database:** Once the database is inserted then the next step is to ensure the performance, the database testing is done by running some queries and comparing with the expected outputs. The DQL[Data Query Language] is used to perform data retrieval, the syntax of the queries used is mentioned below:

- **Syntax to alter table:**

```
SELECT [DISTINCT | ALL] select_expr [, select_expr ...]
  FROM table_name [, table_name ...]
  [WHERE condition]
  [GROUP BY column_name [, column_name ...]]
```

[HAVING condition]

[ORDER BY column_name [ASC | DESC], ...];

6. **Set up security and backup procedures:** After testing the database , one must ensure that that the data needs to be secured, some PL/SQL programs are applied to the database.The PL/SQL is used to create the trigger programs , procedures and functions, the corresponding syntax are given below:

a.Trigger:

```
CREATE [OR REPLACE] TRIGGER trigger_name
  [BEFORE | AFTER] event_name
  ON table_name
  [FOR EACH ROW | FOR EACH STATEMENT]
  [WHEN (condition)]
  BEGIN
    -- trigger body
  END;
```

b.procedure

```
    CREATE [OR REPLACE] PROCEDURE procedure_name (para1,para2..)
IS [AUTHID DEFINER | CURRENT_USER]
        DECLARE statements
BEGIN
        executable statements
END procedure_name;
/
```

c.function

```
CREATE [OR REPLACE] FUNCTION function_name

(parameter 1, parameter2...) RETURN datatype

IS

    Declare Statements

BEGIN

    Executable statements;

    Return (Return Value);

END;
```

7.Monitoring the database: This final step involves to contionusly monitor the database.To ensure that if any problems occurred adjustments need to be done or the way of using database.

4.CONCEPTUAL DATABASE DESIGN:

The conceptual database design is the process of creating a high level representation of the data ,relation and relationships . The major goal is to create an accurate model representing the requirements. This step concludes the further steps. To develop the model , the entities and their attributes need to be identified.

1.ENTITIES AND ATTRIBUTES:

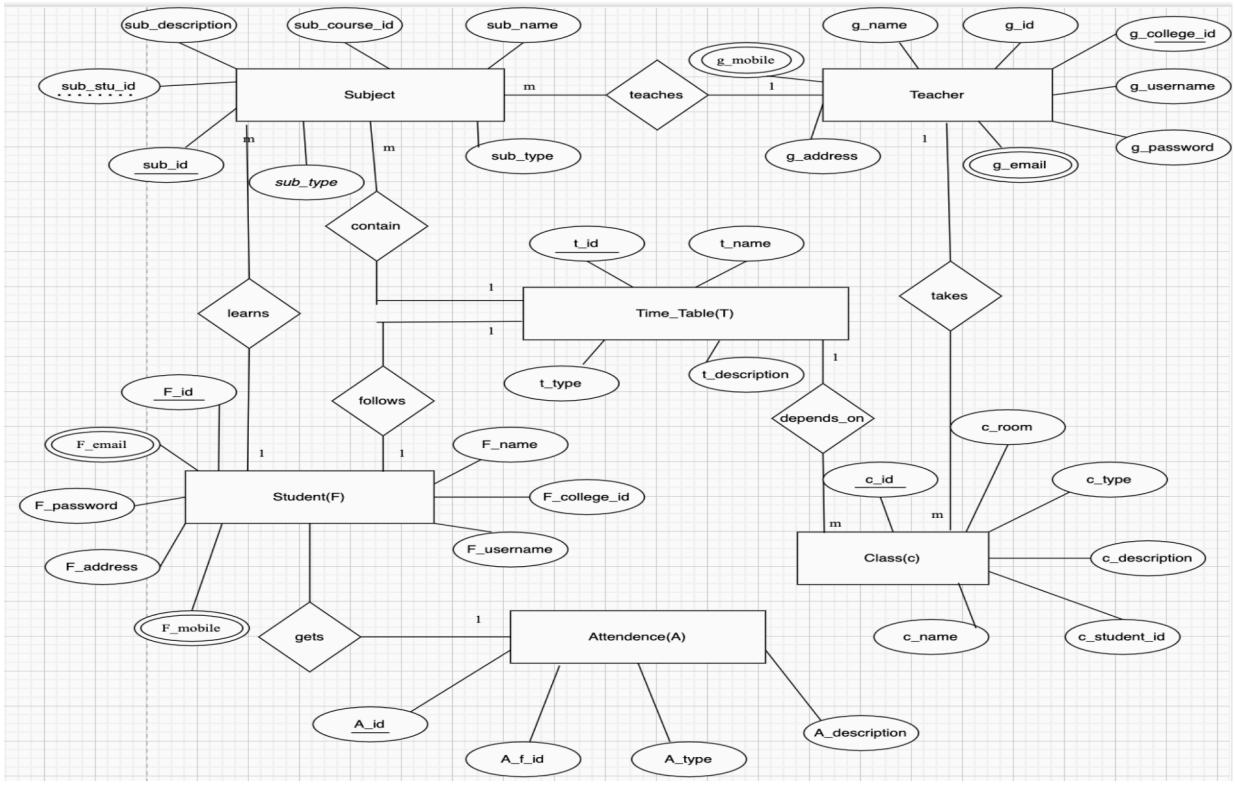
- **Student**
    - Student_id
    - Student_college_id
    - Student_name
    - Student_username
    - Student_address
    - Student_time_table_id
- **Time table**
    - Timetable_id
    - Timetable_name
    - Timetable_type
    - Timetable_description
- **Attendance**
    - Attendance_id
    - Attendance_student_id
    - Attendance_description
    - Attendance_type
- **Class**
    - Class_id
    - Class_student_id
    - Class_name
    - Class_room
    - Class_type
    - Class_decription
- **Subject**
    - Subject_id

- Subject_course_id
- Subject_student_id
- Subject_name
- Subject_type
- Subject_description

- **Teacher**
  - Teacher_id;
  - Teacher_college_id
  - Teacher_name
  - Teacher_mobile
  - Teacher_email
  - Teacher_username

2.ER[ENTITY RELATIONSHIP] MODEL :

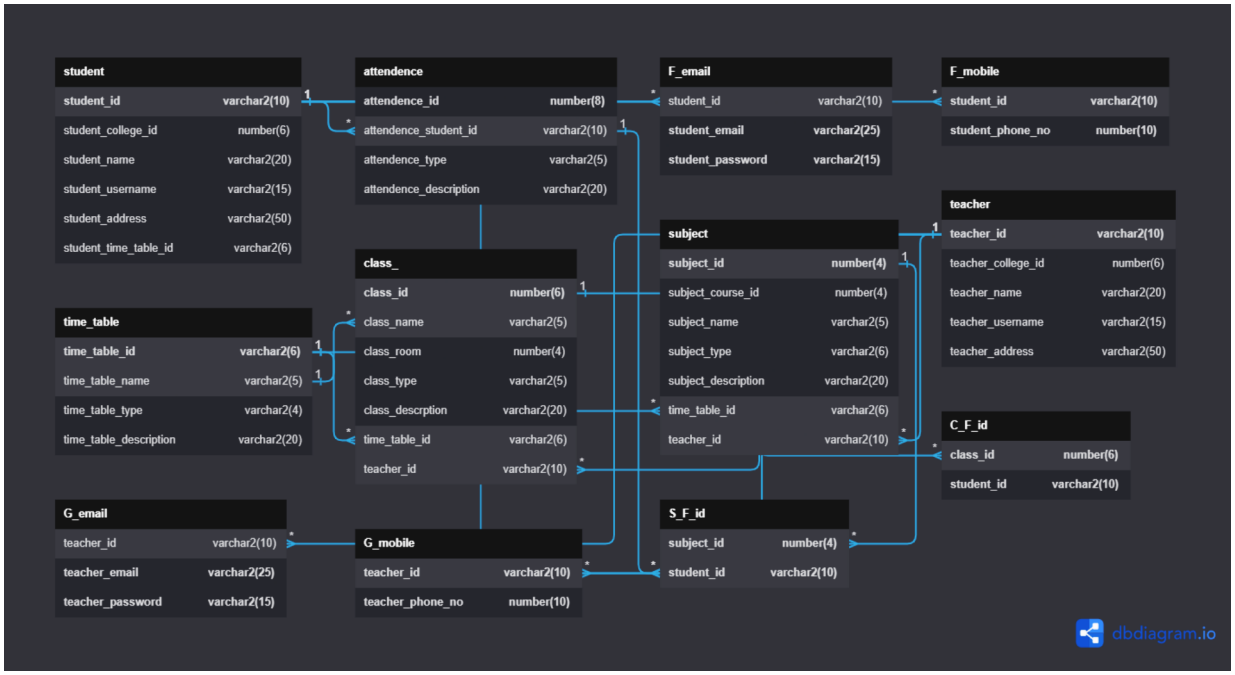
The next step is to design an Entity relationship model, as per the logic and the constraints of the problem. The entities and their relationships are mentioned in the below diagram**:**



The conversion of ER model into relational database design , first we need to convert each entity into tables. Based on the Cardinality constraints new attributes are added to each table , so the existing tables are updated. The multivalued attributes are stored in completely new tables. The table consists of the multivalued attributes along with the primary key attribute from the entity table respective to that attribute.According to the ER model ,there are six multivalued attributes , so there should be six tables needed to be created.

4.DATABASE DIAGRAM:

Here a total of twelve tables are created, each table is connected with foreign key constraint.The foreign key is connected through the primary and unique keys in the entities.



5.LOGICAL DATABASE DESIGN:

1.DDL COMMANDS:

Student table:

create table student(student_id varchar2(10),student_college_id number(6),student_name varchar2(20),student_username varchar2(15),student_address varchar2(50),student_time_table_id varchar2(6),constraint F_pk primary key(student_id),constraint F_uk unique(student_username));

table STUDENT

Column	Null?	Type
STUDENT_ID	NOT NULL	VARCHAR2(10)
STUDENT_COLLEGE_ID	-	NUMBER(6,0)
STUDENT_NAME	-	VARCHAR2(20)
STUDENT_USERNAME	-	VARCHAR2(15)
STUDENT_ADDRESS	-	VARCHAR2(50)
STUDENT_TIME_table_ID	-	VARCHAR2(6)

Attendance:

Create table attendence(attendence_id number(8),attendence_student_id varchar2(10),attendence_type varchar2(5),attendence_description varchar2(20),constraint A_pk primary key(attendence_id));

table ATTENDENCE

Column	Null?	Type
ATTENDENCE_ID	NOT NULL	NUMBER(8,0)
ATTENDENCE_STUDENT_ID	-	VARCHAR2(10)
ATTENDENCE_TYPE	-	VARCHAR2(5)
ATTENDENCE_DESCRIPTION	-	VARCHAR2(20)

F_email:

create table F_email(student_id varchar2(10),student_email varchar2(25),student_password varchar2(15),constraint F_E_pk primary key(student_email,student_password));

table F_EMAIL

Column	Null?	Type
STUDENT_ID	-	VARCHAR2(10)
STUDENT_EMAIL	NOT NULL	VARCHAR2(25)
STUDENT_PASSWORD	NOT NULL	VARCHAR2(15)

F_mobile:

create table F_mobile(student_id varchar2(10),student_phone_no number(10),constraint F_M_pk primary key(student_id,student_phone_no));

Column	Null?	Type
STUDENT_ID	NOT NULL	VARCHAR2(10)
STUDENT_PHONE_NO	NOT NULL	NUMBER(10,0)

Time_table table:

create table time_table(time_table_id varchar2(6),time_table_name varchar2(5),time_table_type varchar2(4),time_table_description varchar2(20),constraint T_pk primary key(time_table_id));

table TIME_table

Column	Null?	Type
TIME_table_ID	NOT NULL	VARCHAR2(6)
TIME_table_NAME	-	VARCHAR2(5)
TIME_table_TYPE	-	VARCHAR2(4)
TIME_table_DESCRIPTION	-	VARCHAR2(20)

Class_table:

create table class_(class_id number(6),class_name varchar2(5),class_room number(4),class_type varchar2(5),class_descrption varchar2(20),time_table_id varchar2(6),teacher_id varchar2(10),constraint C_pk primary key (class_id),constraint C_uk unique (class_name))

table SUBJECT

Column	**Null?**	**Type**
CLASS_ID	NOT NULL	NUMBER(6,0)
CLASS_NAME	-	VARCHAR2(5)
CLASS_ROOM	-	NUMBER(4,0)
CLASS_TYPE	-	VARCHAR2(5)
CLASS_DESCRPTION	-	VARCHAR2(20)
TIME_table_ID	-	VARCHAR2(6)
TEACHER_ID	-	VARCHAR2(10)

Subject table:

create table subject(subject_id number(4),subject_course_id number(4),subject_name varchar2(5),subject_type varchar2(6),subject_description varchar2(20),time_table_id varchar2(6),teacher_id varchar2(10),constraint S_pk primary key (subject_id));

table SUBJECT

Column	Null?	Type
SUBJECT_ID	NOT NULL	NUMBER(4,0)
SUBJECT_COURSE_ID	-	NUMBER(4,0)
SUBJECT_NAME	-	VARCHAR2(5)
SUBJECT_TYPE	-	VARCHAR2(6)
SUBJECT_DESCRIPTION	-	VARCHAR2(20)
TIME_table_ID	-	VARCHAR2(6)
TEACHER_ID	-	VARCHAR2(10)

Teacher table:

create table teacher(teacher_id varchar2(10),teacher_college_id number(6),teacher_name varchar2(20),teacher_username varchar2(15),teacher_address varchar2(50),constraint G_pk primary key(teacher_id),constraint G_uk unique(teacher_username));

Table TEACHER

Column	Null?	Type
TEACHER_ID	-	VARCHAR2(10)
TEACHER_COLLEGE_ID	-	NUMBER(6,0)
TEACHER_NAME	-	VARCHAR2(20)
TEACHER_USERNAME	-	VARCHAR2(15)

TEACHER_ADDRESS	-	VARCHAR2(50)

G_email:

create table G_email(teacher_id varchar2(10),teacher_email varchar2(25),teacher_password varchar2(15),constraint G_E_pk primary key(teacher_email,teacher_password));

Column	**Null?**	**Type**
TEACHER_ID	-	VARCHAR2(10)
TEACHER_EMAIL	NOT NULL	VARCHAR2(25)
TEACHER_PASSWORD	NOT NULL	VARCHAR2(15)

G_mobile:

create table G_mobile(teacher_id varchar2(10),teacher_phone_no number(10),constraint G_M_pk primary key(teacher_id,teacher_phone_no));

Column	**Null?**	**Type**
TEACHER_ID	NOT NULL	VARCHAR2(10)
TEACHER_PHONE_NO	NOT NULL	NUMBER(10,0)

S_F_id:

create table S_F_id(subject_id number(4),student_id varchar2(10),constraint S_F_pk primary key (subject_id,student_id));

Column	Null?	Type
SUBJECT_ID	NOT NULL	NUMBER(4,0)
STUDENT_ID	NOT NULL	VARCHAR2(10)

C_F_id:

create table C_F_id(class_id number(6),student_id varchar2(10),constraint C_F_pk primary key (class_id,student_id));

Column	**Null?**	**Type**
CLASS_ID	NOT NULL	NUMBER(6,0)
STUDENT_ID	NOT NULL	VARCHAR2(10)

2.CONSTRAINTS:

- alter table attendence add(constraint A_F_fk foreign key(attendence_student_id) references student(student_id));
- alter table F_email add constraint F_email_F_fk foreign key(student_id) references student(student_id);
- alter table F_mobile add constraint F_mobile_F_fk foreign key(student_id) references student(student_id);
- alter table S_F_id add constraint S_F_id_S_fk foreign key(subject_id) references subject(subject_id);
- alter table S_F_id add constraint S_F_id_F_fk foreign key(student_id) references student(student_id);
- alter table subject add constraint S_T_fk foreign key(time_table_id) references time_table(time_table_id);
- alter table subject add constraint S_G_fk foreign key(teacher_id) references teacher(teacher_id);
- alter table C_F_id add constraint C_F_id_C_fk foreign key(class_id) references class_(class_id);
- alter table class_ add constraint C_T_fk foreign key(time_table_id) references time_table(time_table_id);
- alter table class_ add constraint C_G_fk foreign key(teacher_id) references teacher(teacher_id);
- alter table G_email add constraint G_email_G_fk foreign key(teacher_id) references teacher(teacher_id);
- alter table G_mobile add constraint G_mobile_G_fk foreign key(teacher_id) references teacher(teacher_id);
- alter table time_table add constraint T_uk unique(time_table_name);
- alter table class_ add constraint C_T_ufk foreign key(class_name) references time_table(time_table_name);
- alter table attendence add constraint A_uk unique(attendence_student_id);
- alter table S_F_id add constraint S_F_id_A foreign key(student_id) references attendence(attendence_student_id);

6.DML COMMANDS:

student table:

- insert into student values('212141A01',151671,'Dharma','Ala@143','sheikpet,lakdikapool,hyderabad','CSE');
- insert into student values('21241A02',151671,'Manga sri','MD@314','sheikpet,lakdikapool,hyderabad','CSE');
- insert into student values('21241A03',151671,'Kaveri','KD@314','Musherrabad,hyderabad','ECE');
- insert into student values('21241A04',151671,'Lailavathi','LD@314','Malakpet,hyderabad','ECE');
- insert into student values('21241A05',257131,'Padmavathi','PD@314','Amberpet,hyderabad','IT');
- insert into student values('21241A06',257231,'Saikiran','PSK@143','khairatabad,hyderabad','EEE');
- insert into student values('21241A07',423110,'Kruthika','KSK@341','Sanath nagar,hyderabad','IT');
- insert into student values('21241A08',483261,'Krupa','RSK@341','Karwan,hyderabad','EEE');
- insert into student values('21241A09',257231,'Swathi','SSK@341','Nampally,hyderabad','CSE');
- insert into student values('21241A10',483161,'Raghavendra','RD@143','Goshhamahal,hyderabad','CSE');
- insert into student values('21241A11',483261,'Dathu Sai','DD@143','Charminar,hyderabad','CSE');
- insert into student values('21241A12',483261,'Samad','SM@143','Chandrayangutta,hyderabad','EEE');
- insert into student values('21241A13',483261,'Ahmed','AM@123','Yakutpura,hyderabad','IT');
- insert into student values('21241A14',118921,'Jones Richard','JRH@821','Bahadurpura,hyderabad','IT');
- insert into student values('21241A15',118921,'Chan lee','CL@111','parade ground,secunderabad','CSE');
- insert into student values('212141A05',483261,'Krupa','RSKP@341','Karwan,hyderabad','EEE');
- insert into student values('212141A06',483261,'Kranthi','kra@341','Karwan,hyderabad','EEE');
- insert into student values('21241A01',151671,'sriram','SR@678','malkajgiri,secunderabad','CSE');

STUDENT_ID	STUDENT_COLLEGE_ID	STUDENT_NAME	STUDENT_USERNAME	STUDENT_ADDRESS	STUDENT_TIME_table_ID
212141A01	151671	Dharma	Ala@143	sheikpet,lakdikapool,hyderabad	CSE
21241A02	151671	Manga sri	MD@314	sheikpet,lakdikapool,hyderabad	CSE
21241A03	151671	Kaveri	KD@314	Musherrabad,hyderabad	ECE
21241A04	151671	Lailavathi	LD@314	Malakpet,hyderabad	ECE

21241A05	257131	Padmavathi	PD@314	Amberpet,hyderabad	IT
21241A06	257231	Sai kiran	PSK@143	khairatabad,hyderabad	EEE
21241A07	423110	Kruthika	KSK@341	Sanath nagar,hyderabad	IT
21241A08	483261	Krupa	RSK@341	Karwan,hyderabad	EEE
21241A09	257231	Swathi	SSK@341	Nampally,hyderabad	CSE
21241A10	483161	Raghavendra	RD@143	Goshhamahal,hyderabad	CSE
21241A11	483261	Dathu Sai	DD@143	Charminar,hyderabad	CSE
21241A12	483261	Samad	SM@143	Chandrayangutta,hyderabad	EEE
21241A13	483261	Ahmed	AM@123	Yakutpura,hyderabad	IT
21241A14	118921	Jones Richard	JRH@821	Bahadurpura,hyderabad	IT
21241A15	118921	Chan lee	CL@111	parade ground,secunderabad	CSE
212141A05	483261	Krupa	RSKP@341	Karwan,hyderabad	EEE
212141A06	483261	Kranthi	kra@341	Karwan,hyderabad	EEE
21241A01	151671	sriram	SR@678	malkajgiri,secunderabad	CSE

time_table:

- insert into time_table values('51122B','T_ECE','CL_T','time_t of ece staff');
- insert into time_table values('52761L','T_EEE','CL_T','time_t of eee staff');
- insert into time_table values('53842C','T_IT','CL_T','time_t of it staff');
- insert into time_table values('77777H','HOD','HD_T','time_t of hod');
- insert into time_table values('41212A','CSE','ST_T','time_t of cse stu');
- insert into time_table values('43212B','ECE','ST_T','time_t of ece stu');
- insert into time_table values('48262L','EEE','ST_T','time_t of eee stu');
- insert into time_table values('49124B','IT','ST_T','time_t of it stu');
- insert into time_table values('50124A','T_CSE','CL_T','time_t of cse staff');
- insert into time_table values('58262L','CSM','ST_T','time_t of csm stu');

TIME_table_ID	TIME_table_NAME	TIME_table_TYPE	TIME_table_DESCRIPTION
51122B	T_ECE	CL_T	time_t of ece staff
52761L	T_EEE	CL_T	time_t of eee staff
53842C	T_IT	CL_T	time_t of it staff
77777H	HOD	HD_T	time_t of hod
41212A	CSE	ST_T	time_t of cse stu
43212B	ECE	ST_T	time_t of ece stu
48262L	EEE	ST_T	time_t of eee stu
49124B	IT	ST_T	time_t of it stu
50124A	T_CSE	CL_T	time_t of cse staff
58262L	CSM	ST_T	time_t of csm stu

Attendence table:

- insert into attendence values(12345678,'212141A01','u_exc','unexcused student');
- insert into attendence values(62317137,'21241A02','u_exc','unexcused student');
- insert into attendence values(86178177,'21241A03','exc','excused student');
- insert into attendence values(13987989,'21241A04','exc','excused student ');
- insert into attendence values(83427783,'21241A05','u_exc','unexcused student ');
- insert into attendence values(73279823,'21241A06','exc','excused student ');
- insert into attendence values(82387231,'21241A07','u_exc','unexcused student ');
- insert into attendence values(23623177,'21241A08','u_exc','unexcused student');
- insert into attendence values(18278723,'21241A09','u_exc','unexcused student');
- insert into attendence values(28327663,'21241A10','u_exc','unexcused student');
- insert into attendence values(23719873,'21241A11','u_exc','unexcused student');

- insert into attendence values(92319889,'21241A12','u_exc','unexcused student');
- insert into attendence values(23893288,'21241A13','u_exc','unexcused student');
- insert into attendence values(87213878,'21241A14','u_exc','unexcused student');
- insert into attendence values(92139899,'21241A15','exc','excused student');
- insert into attendence values(87654321,'21241A01','u_exc','unexcused student');

ATTENDENCE_ID	ATTENDENCE_STUDENT_ID	ATTENDENCE_TYPE	ATTENDENCE_DESCRIPTION
12345678	212141A01	u_exc	unexcused student
62317137	21241A02	u_exc	unexcused student
86178177	21241A03	exc	excused student
13987989	21241A04	exc	excused student
83427783	21241A05	u_exc	unexcused student
73279823	21241A06	exc	excused student
82387231	21241A07	u_exc	unexcused student
23623177	21241A08	u_exc	unexcused student
18278723	21241A09	u_exc	unexcused student
28327663	21241A10	u_exc	unexcused student
23719873	21241A11	u_exc	unexcused student
92319889	21241A12	u_exc	unexcused student
23893288	21241A13	u_exc	unexcused student

87213878	21241A14	u_exc	unexcused student
92139899	21241A15	exc	excused student
87654321	21241A01	u_exc	unexcused student

Teacher table:

- The teacher_table is inserted with 9 records the insert statements are given below:
- insert into teacher values('T28718','87889','Samitha','Sam@889','Nandi hills,bangalore');
- insert into teacher values('T28719','21982','Srihan','Sri@982','kalina,mumbai');
- insert into teacher values('T28778','21892','jhansi','jhan@892','hussain sagar,secunderabad');
- insert into teacher values('T28700','21819','bruhati','bru@819','Kukatpally,hyderabad');
- insert into teacher values('T28799','87889','Revathi','Rev@889','Nandi hills,bangalore');
- insert into teacher values('T28792','21912','Revanth','Rev@912','Medchal,Hyderabad');
- insert into teacher values('T28732','22344','Jennifer','Jen@344','Lenin tomb ,moscow');
- insert into teacher values('T28793','78372','Brian','Bri@372','Mayakoba,Mexico');
- insert into teacher values('T28779','87889','jyothika','jyo@098','nizampet,hyderabad');

TEACHER_ID	TEACHER_COLLEGE_ID	TEACHER_NAME	TEACHER_USERNAME	TEACHER_ADDRESS
T28718	87889	Samitha	Sam@889	Nandi hills,bangalore
T28719	21982	Srihan	Sri@982	kalina,mumbai
T28778	21892	jhansi	jhan@892	hussain sagar,secunderabad
T28700	21819	bruhati	bru@819	Kukatpally,hyderabad
T28799	87889	Revathi	Rev@889	Nandi hills,banglore

T28792	21912	Revanth	Rev@912	Medchal,Hyderabad
T28732	22344	Jennifer	Jen@344	Lenin tomb ,moscow
T28793	78372	Brian	Bri@372	Mayakoba,Mexico
T28779	87889	jyothika	jyo@098	nizampet,hyderabad

G_email table:

The G_email table is inserted with 7 records the insert statements are given below:

- insert into G_email values('T28718','samitha@gmail.com','9292388231');
- insert into G_email values('T28718','samitha71@gmail.com','c27c7ihew7h2');
- insert into G_email values('T28719','srihan@gmail.com','4j832j8jd');
- insert into G_email values('T28778','Jhansi@gmail.com','8j349fru9');
- insert into G_email values('T28700','bruhati@gmail.com','x1e7yxeb');
- insert into G_email values('T28792','revanth@gmail.com','76g77vt6ytv');
- insert into G_email values('T28732','jennifer@gmail.com','67gbb67o7y');

TEACHER_ID	TEACHER_EMAIL	TEACHER_PASSWORD
T28718	samitha@gmail.com	9292388231
T28718	samitha71@gmail.com	c27c7ihew7h2
T28719	srihan@gmail.com	4j832j8jd
T28778	Jhansi@gmail.com	8j349fru9
T28700	bruhati@gmail.com	x1e7yxeb
T28792	revanth@gmail.com	76g77vt6ytv

T28732	jennifer@gmail.com	67gbb67o7y

F_email table:

The F_email table is inserted with 5 records the insert statements are given below:

- insert into F_email values('21241A02','manga@gamil.com','2x817809');
- insert into F_email values('21241A03','kaveri@gamil.com','3298932812');
- insert into F_email values('21241A04','lailavathi@gamil.com','b7x1387s');
- insert into F_email values('21241A05','padmavathi@gamil.com','b7218y8uq');
- insert into F_email values('21241A06','saikiran@gamil.com','7b8128812');

STUDENT_ID	STUDENT_EMAIL	STUDENT_PASSWORD
21241A03	kaveri@gamil.com	3298932812
21241A04	lailavathi@gamil.com	b7x1387s
21241A05	padmavathi@gamil.com	b7218y8uq
21241A06	saikiran@gamil.com	7b8128812
21241A02	manga@gamil.com	2x817809

G_mobile table:

The G_mobile table is inserted with 10 records the insert statements are given below:

- insert into G_mobile values('T28718',9898912910);
- insert into G_mobile values('T28718',8127818910);
- insert into G_mobile values('T28719',8879909009);
- insert into G_mobile values('T28719',7897899009);
- insert into G_mobile values('T28778',8909038989);
- insert into G_mobile values('T28700',2392318921);

TEACHER_ID	TEACHER_PHONE_NO
T28700	2392318921
T28718	8127818910
T28718	9898912910
T28719	7897899009
T28719	8879909009
T28778	8909038989

F_mobile table:

The F_mobile table is inserted with 5 records the insert statements are given below:

- insert into F_mobile values('212141A01',9280287657);
- insert into F_mobile values('212141A01',7129891910);
- insert into F_mobile values('212141A05',1089090389);
- insert into F_mobile values('212141A05',2139012992);
- insert into F_mobile values('212141A06',9092198892);

STUDENT_ID	STUDENT_PHONE_NO
212141A01	7129891910
212141A01	9280287657
212141A05	1089090389
212141A05	2139012992
212141A06	9092198892

Subject table:

The subject_table is inserted with 5 records the insert statements are given below:

- insert into subject values(1231,4124,'DBMS','theory','DBMS THEORY INFO.','41212A','T28718');
- insert into subject values(1230,4124,'P&S','theory','P&S THEORY INFO.','41212A','T28719');
- insert into subject values(1232,4120,'ELCS','lang','English language','43212B','T28778');
- insert into subject values(1233,4120,'FSD','theory','FSD THEORY INFO.','43212B','T28700');
- insert into subject values(1235,4123,'BEE','theory','BEE THEORY INFO.','50124A','T28792');

SUBJECT_ID	SUBJECT_COURSE_ID	SUBJECT_NAME	SUBJECT_TYPE	SUBJECT_DESCRIPTION	TIME_table_ID	TEACHER_ID
1231	4124	DBMS	theory	DBMS THEORY INFO.	41212A	T28718
1230	4124	P&S	theory	P&S THEORY INFO.	41212A	T28719
1232	4120	ELCS	lang	English language	43212B	T28778
1233	4120	FSD	theory	FSD THEORY INFO.	43212B	T28700
1235	4123	BEE	theory	BEE THEORY INFO.	50124A	T28792

Class_ table:

The class_ table is inserted with 4 records the insert statements are given below:

- insert into class_ values(12715,'IT',3403,'sub','for IT theory class.','49124B','T28718');
- insert into class_ values(12712,'ECE',4403,'sub','for ece theory class','43212B','T28719');
- insert into class_ values(12713,'EEE',2403,'sub','for eee theory class','48262L','T28778');
- insert into class_ values(12714,'CSE',1403,'sub','for cse theory class','41212A','T28700');

CLASS_ID	CLASS_ NAME	CLASS_ ROOM	CLASS_TYPE	CLASS_DESCRPTION	TIME_table_ID	TEACHER_I D
12715	IT	3403	sub	for IT theory class.	49124B	T28718
12712	ECE	4403	sub	for ece theory class	43212B	T28719
12713	EEE	2403	sub	for eee theory class	48262L	T28778
12714	CSE	1403	sub	for cse theory class	41212A	T28700

S_F_id table:

The S_F_id table is inserted with 4 records the insert statements are given below:

- insert into S_F_id values(1231,'212141A01');
- insert into S_F_id values(1231,'21241A02');
- insert into S_F_id values(1231,'21241A10');
- insert into S_F_id values(1231,'21241A11');

SUBJECT_ID	STUDENT_ID
1231	212141A01
1231	21241A02
1231	21241A10
1231	21241A11

C_F_id table:

The C_F_id table is inserted with 4 records the insert statements are given below:

- insert into C_F_id values(12715,'212141A01');
- insert into C_F_id values(12715,'21241A02');
- insert into C_F_id values(12715,'21241A10');
- insert into C_F_id values(12715,'21241A11');

CLASS_ID	STUDENT_ID
12715	21241A02
12715	21241A10
12715	21241A11
12715	212141A01

7.DQL COMMANDS

1)Retrive data of student with roll no.=1231

select * from subject where subject_id=1231;

OUTPUT:

```
SUBJECT_ID SUBJECT_COURSE_ID SUBJE SUBJEC SUBJECT_DESCRIPTION  TIME_T TEACHE
R_ID
---------- ----------------- ----- ------ -------------------- ------ ------
----
      1231              4124 DBMS  theory DBMS THEORY INFO.    41212A T28718
```

2.Count total number of classes follows the time table whose id ends with B.

select count(*) from class_ where time_table_id like '%B';

OUTPUT:

```
  COUNT(*)
----------
         2
```

3.Display student name, Student phone number, Student user name of the student.

select s.student_id,s.student_username,g.student_phone_no from student s cross join F_mobile g where s.student_id=g.student_id;

OUTPUT:

```
STUDENT_ID  STUDENT_USERNAM  STUDENT_PHONE_NO
----------  ---------------  ----------------
212141A01   Ala@143                7129891910
212141A01   Ala@143                9280287657
212141A05   RSKP@341               1089090389
212141A05   RSKP@341               2139012992
212141A06   kra@341                9092198892
```

4.Display student name, student id who are excused.

select student_id,student_name from student where student_id in (select attendence_student_id from attendence where attendence_type = 'exc');

OUTPUT:

```
STUDENT_ID STUDENT_NAME
---------- --------------------
21241A03   Kaveri
21241A04   Lailavathi
21241A06   Sai kiran
21241A15   Chan lee
```

5.Display the first three characters of all students in the student table.

select substr(student_name,0,3) from student;

OUTPUT:

```
SUBSTR(STUDE
------------
Dha
Man
Kav
Lai
Pad
Sai
Kru
Kru
Swa
Rag
Dat

SUBSTR(STUDE
------------
Sam
Ahm
Jon
Cha
Kru
Kra
sri
Man

19 rows selected.
```

6.Create a view that displays the student id, student name who belong to cse department.

create or replace view stu_view as select student_id,student_name from student where student_time_table_id='CSE';

OUTPUT:

```
View created.

SQL> select * from stu_view;

STUDENT_ID STUDENT_NAME
---------- --------------------
212141A01  Dharma
21241A02   Manga sri
21241A09   Swathi
21241A10   Raghavendra
21241A11   Dathu Sai
21241A15   Chan lee
21241A01   sriram

7 rows selected.
```

8.PL/SQL PROGRAMS

1.Trigger to store the information of logon date and time

PROGRAM

```
CREATE TABLE hr_event_audit(logon_date DATE,logon_time
VARCHAR2(15));

CREATE OR REPLACE TRIGGER hr_lgon_audit
AFTER LOGON ON SCHEMA
BEGIN
        INSERT INTO hr_event_audit VALUES(sysdate,to_char(sysdate,'hh24:mi:ss'));
COMMIT;
END;
/
```

OUTPUT:

```
SQL> @D:\Btech_2.1\DBMS\Class\Assignment_1\project_pl_ddl_trigger.sql;


Table created.


Trigger created.
```

```
Enter user-name: project
Enter password:
Connected.
SQL> select * from hr_event_audit;

LOGON_DAT LOGON_TIME
--------- ---------------
27-JAN-23 21:50:20
```

2. Trigger to store backup data of student table. In this program the backup data is student_id.

PROGRAM

```
CREATE OR REPLACE TRIGGER stu_backup
BEFORE INSERT OR UPDATE OR DELETE ON student
FOR EACH ROW
BEGIN
        IF INSERTING THEN
                INSERT INTO student_backup (student_id) values (:NEW.student_id);
        ELSIF DELETING THEN
                DELETE FROM student_backup WHERE student_id=:OLD.student_id;
        ELSIF UPDATING THEN
                UPDATE student_backup SET student_id=:NEW.student_id WHERE student_id=:OLD.student_id;
        END IF;
END;
/
```

OUTPUT:

```
SQL> set serveroutput on;
SQL> @D:\Btech_2.1\DBMS\Class\Assignment_1\project_dml_trigger.sql
 13  /

Trigger created.

SQL> insert into student (student_id,student_name) values (90122,'Manohar');

1 row created.
```

```
SQL> select student_id from student_backup;

STUDENT_ID
----------
212141A01
21241A02
21241A03
21241A04
21241A05
21241A06
21241A07
21241A08
21241A09
21241A10
21241A11

STUDENT_ID
----------
21241A12
21241A13
21241A14
21241A15
212141A05
212141A06
21241A01
90122

19 rows selected.
```

```
SQL> select student_id from student;

STUDENT_ID
----------
212141A01
212141A05
212141A06
21241A01
21241A02
21241A03
21241A04
21241A05
21241A06
21241A07
21241A08

STUDENT_ID
----------
21241A09
21241A10
21241A11
21241A12
21241A13
21241A14
21241A15
90122

19 rows selected.
```

3.Procedure to know the students who need to pay condonation fees.

PROGRAM

```
create or replace procedure p_at_few
is
        cursor c_fee is select *from at_fees;
        v_i at_fees.attendence_type%type;
begin
        for l_index IN c_fee
        loop
                if l_index.attendence_type like '%u_exc%' then
                        dbms_output.put_line('Student '||l_index.attendence_student_id||' need to pay condonation fee 500/-');
                end if;
        end loop;
End;
/
```

OUTPUT:

```
SQL> @D:\Btech_2.1\DBMS\Class\Assignment_1\project_pl_sql_procedure.sql
  14  /

Procedure created.
```

```
SQL> set serveroutput on;
SQL> exec p_at_few;
Student 212141A01 need to pay condonation fee 500/-
Student 21241A02 need to pay condonation fee 500/-
Student 21241A05 need to pay condonation fee 500/-
Student 21241A07 need to pay condonation fee 500/-
Student 21241A08 need to pay condonation fee 500/-
Student 21241A09 need to pay condonation fee 500/-
Student 21241A10 need to pay condonation fee 500/-
Student 21241A11 need to pay condonation fee 500/-
Student 21241A12 need to pay condonation fee 500/-
Student 21241A13 need to pay condonation fee 500/-
Student 21241A14 need to pay condonation fee 500/-
Student 21241A01 need to pay condonation fee 500/-

PL/SQL procedure successfully completed.
```

9.CONCLUSION

In conclusion, the process of testing and creating a database involves several steps, in this journey all these steps are mostly covered. The initial design and planning of the database is crucial to its success. But one more thing, one must ensure the security of the database. The database administrator monitored and maintained this database. By this we can adjust to prevent any logical and semantic exceptions occurring in the design. This way one can increase efficiency of the design.

10.REFERENCES

S.NO	REFERENCES
1	“Database Management Systems”, Raghurama Krishnan.
2	“Database System Concepts”, Silberschatz, Korth.
3	DBMS: Oracle 10g, live sql Oracle Live SQL .
4	SQL FILE Format: Notepad
5	ER-MODEL: Untitled Diagram - diagrams.net
6	DATABASE-DIAGRAM : dbdiagram.io - Database Relationship Diagrams Design Tool
7	PL/SQL: https://youtube.com/watch?v=rbarR4_gaH8&feature=shares