SCHOOL MANAGEMENT SYSTEM

WORK BY:

1.LANKA JASWANTH (19BIT0061)
2.YESWIN CHOWDARY (19BIT0134)
3.B K ANIRUDH (19BIT0348)

GUIDED BY:

PROF.BIMAL KUMAR RAY

INTRODUCTION

The education system forms the backbone of every nation. Hence it is important and necessary to provide a strong educational foundation to the young generation to ensure the development of open-minded citizens securing the future for everyone. Advanced technology today can play a crucial role in streamlining the education-related processes to promote solidarity among students, teachers, parents, and the school staff.

Existing System

In the current system, we need to keep the number of records related to students and want to enter the marks and records of the students manually. In this system teacher or school, authority views the marks and records of the student and keeps track of the student manually. School authority needs to keep track of all the members of the school including students. This is very time consuming and has a lot of paperwork.

Need for the System

In our proposed system, We have the provision for adding the details pf the student. The overhead of the school authorities and teachers becomes less. Another advantage of the system is that it is very easy to edit the details of the student and delete the records if necessary.

Advantages

- Fast access to the database
- Fewer errors
- More storage capacity
- Student and faculty friendly
- Cost convenient
- Less manpower

The system will be used for conducting day to day teaching activities such as assigning homework to conduct exams along with management tasks such as checking the fees status of the student, keeping a record of the number of students enrolled in a class and subject. It is not easy to do this process manually because it would become very hectic. Hence it is recommended to automate the process by developing the relevant software as the world is moving from manual working to information and technology era where computerization becomes important in all parts of life.

Data Requirements

The school management system has a set of data requirements. The following are the entity types that will be used in our school management system.

1.ADMIN_DPT:

This entity consists of A_ID which is unique to each administrator be it a principal or supervisor. The other attributes are Name, Salary, Designation, Qualification, Email, Phone_no.

2.CLASS:

CLASS as an entity consists of attributes such as Std_Sec which is unique to each class, Total number of students.

3.TEACHER:

TEACHER as an entity consists of attributes T_ID which is unique to each teacher, Qualification, Salary, Name, Phone_number, Class_taken.

4.EXAM:

EXAM as an entity consists of attributes Exam_code which is unique to every exam conducted, Exam_sub, Exam_date, Exam_qpaper.

5.HOMEWORK:

HOMEWORK as an entity consists of HW_ID which is used to uniquely identify homework given by faculty, Sub_assign, Due_date, Numer_of_students.

6.MARKS:

MARKS as an entity type consists of Paper_no as identifying key, Marks_obt, Marks_tot, Subject.

7.SUBJECT:

SUBJECT entity type has attributes Sub_code as identifying key, Name, ,Total_stu_enrolled as other attributes.

8.STUDENT:

STUDENT as an entity type consists of attributes such as Roll_no unique to every student, Section, Name, Class, Fees.

9.PARENT:

PARENT is a **Weak entity of Student** consisting of attributes such as Roll_no which is unique to every parent, name, Phone_num.

Relationships

- Parent GUARDS Student (1- 1)
  Each parent guards only one student and Parent exist only if he has a student in the school. Hence participation of Parent in GUARDS relation is **Total Participation.(Total participation of Parent and Partial participation of student in this relationship)**
- Teahcer GIVES Marks(1-N)
  Each Teacher gives many marks to many students in the miniworld.Hence the relationships becomes (1-N) relation ship.**(Partial Participation of Teacher and Marks in Moniters relationship)**
- Teacher KEEPS Exam (1-1)
  Each Teacher has to post only 1 test regarding his subject. So, the relationship is a 1-1 relationship. **(Partial Participation of both Teacher and Exam)**
- Teacher GIVES Homework(1-N)
  A Teacher may give may Homeworks regarding his subject. So, the relationship becomes a 1-N relationship. **(Partial Participation of both Teacher and Homework)**
- Teacher SPECIALIZE Subject(1-1)
  A Teacher may teach only 1 subject. Hence, the relationship becomes a 1-1 relationship. **(Partial Participation of both Teacher and Subject)**
- Teacher TEACHES Class(1-N)
  A Teacher may have many classes in a day. Hence, the relationship is a 1-N relationship. **(Partial Participation of both Teacher and Class)**
- Admin_dpt MANAGES Class(1-N)
  An administrator(Principal or Supervisor) have to manage all the class. Hence, the relationship is a 1-N relationship. **(Partial Participation of both Admin_dpt and Class)**
- Admin_dpt SUPERVISES Teacher(1-N)
  An administrator(Principal or Supervisor) has to manage all the teachers of the school. Hence, the relationship is a 1-N relationship. **(Partial Participation of both Admin_dpt and Teacher)**

Functional Requirements

Removal of Data

There are certain scenarios when the data stored with the School Management System has to be deleted. This could arise from a situation such as a student, faculty, admin leaving his/her job. In such cases, the data stored about the individual must be deleted to prevent unnecessary storage of data and to ensure the security of our data. Scenarios demanding deletion/removal of data are discussed below:

Case 1:
A situation may arise where the number of students enrolled in a given class is less than the expected number, say 10. In such a case, the class of that particular subject is deleted.

Case 2:
If in case a teacher wants to delete homework assigned due to some reasons.

Case 3:
There are certain situations such as when the students want to leave school. In such cases, we have to remove his/her account as well as all the data associated with it.

Case 4:
If in case a course does not exist anymore i.e. the school no longer offers a particular course then we have to delete all information regarding that subject.

Modification of Data

School management system a highly efficient system. Therefore, it has to be updated regularly to maintain the correct set of data. And there are situations where the details of an individual changes

Case 1:
Change of Contact Number In case, a student's parent has a change of contact number which has been reported. Then, the admin, on request from the user, changes the student's parent's number
.
Case 2:
In case the marks entered by a teacher has to be changed then only the particular subject teacher is allowed to change the marks

Case 3:
There could be a case where the question paper is to be changed may be due to a moderation

Case 4:
In case a teacher wishes to change the deadline for homework. This function can only be performed by the subject teacher.

Selection of Data

Data retrieval is an operation performed by the system on request by the user or the admin to generate the output which is asked by the user. This could include arithmetic operations such as average or could be simply filtering something out of the database

Case 1:
print out all the students who have secured first division in a particular subject. (first division is marked greater than 95%)

Case 2:
Similarly, we can search for the admins who own an engineering degree.

Case 3:
our System can print the fee status of a student which could be paid. This would be highly helpful in knowing the students whose fees are still pending.

Case 4:
Similarly, we can also find students who have paid the fees.

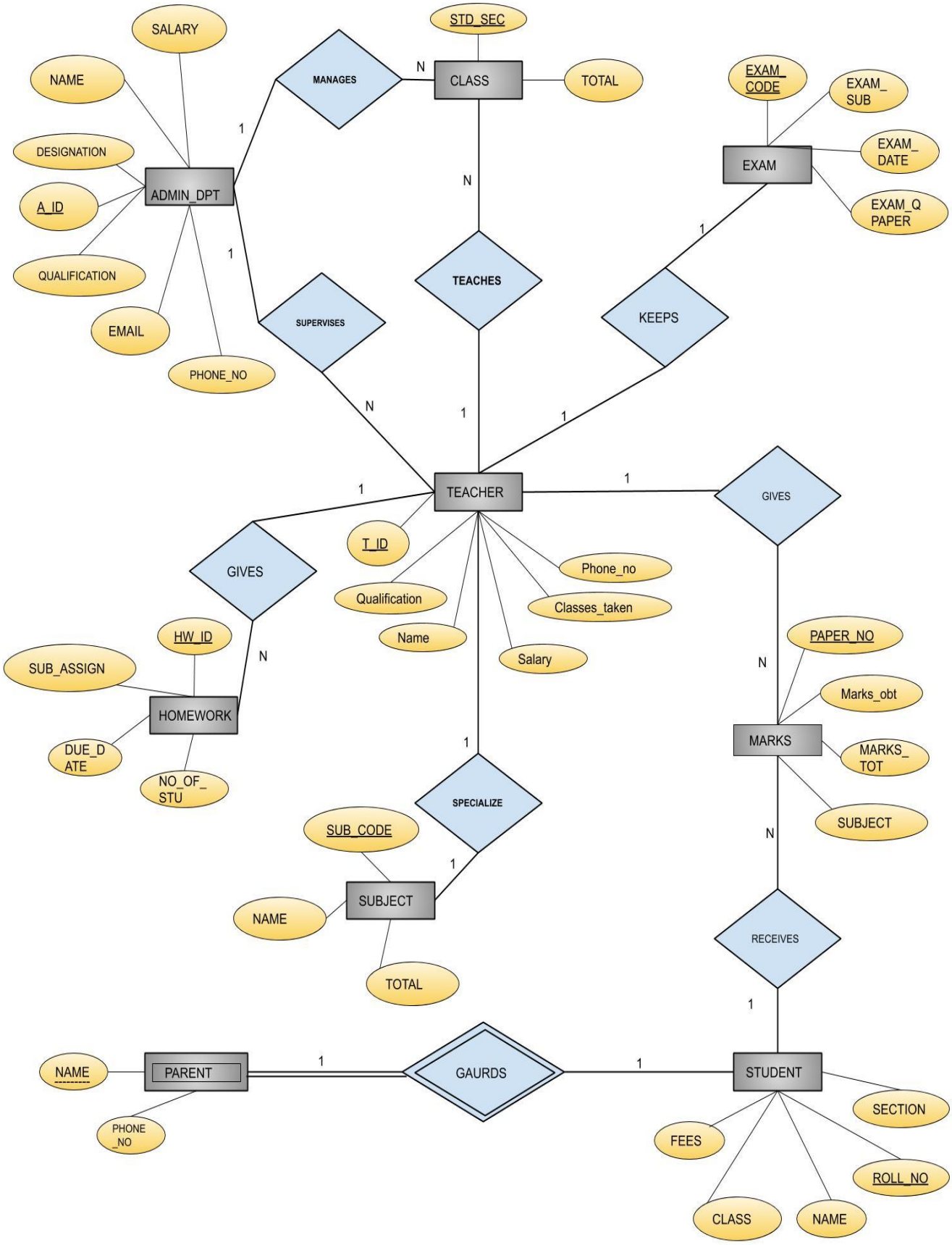
Case 5:
Print students who have received more than 40 in one or more subjects.

Case 6:
using teacher id we could see permanent teachers in the school

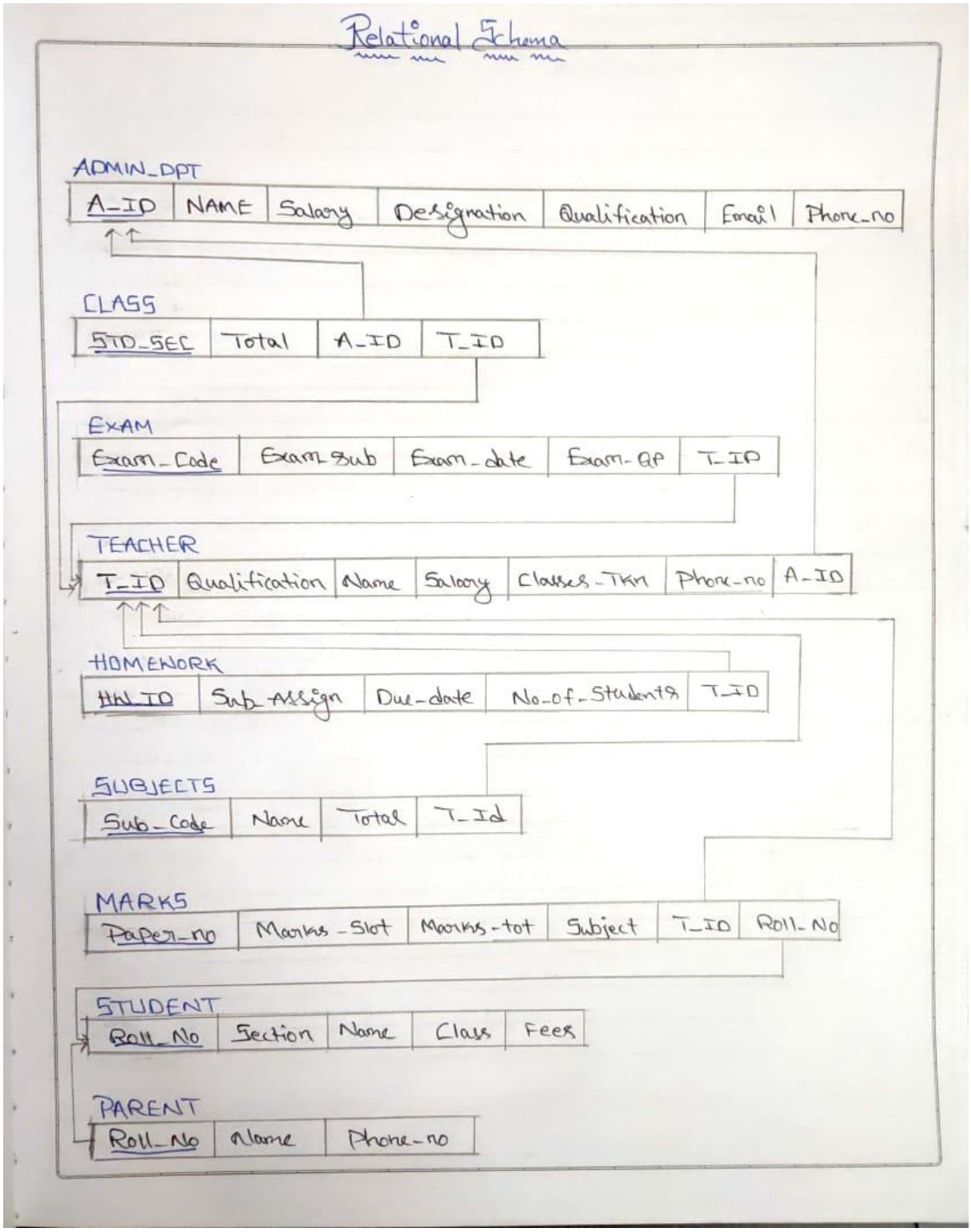
Case 7:
Teachers supervised by a specific coordinator

Case 8:
We can find/filter the teachers who own a Doctorate.

ER DIAGRAM

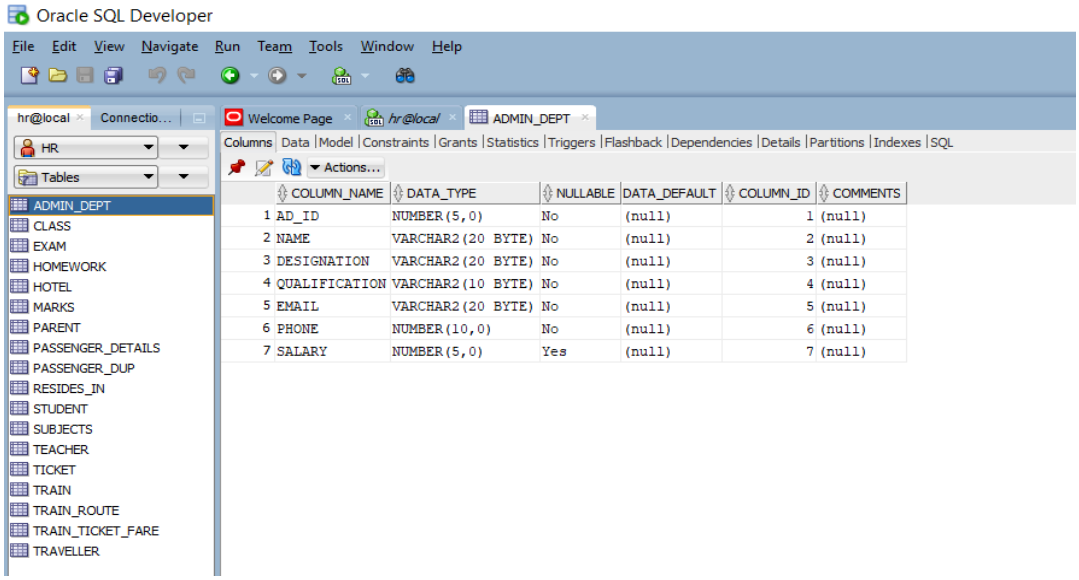

SALARY
NAME
DESIGNATION
A_ID
QUALIFICATION
EMAIL
PHONE_NO
ADMIN_DPT
MANAGES
STD_SEC
CLASS
TOTAL
EXAM_ CODE
EXAM_ SUB
EXAM_ DATE
EXAM_Q PAPER
EXAM
TEACHES
SUPERVISES
KEEPS
TEACHER
T_ID
Qualification
Name
Phone_no
Classes_taken
Salary
GIVES
GIVES
HW_ID
SUB_ASSIGN
HOMEWORK
DUE_D ATE
NO_OF_ STU
SPECIALIZE
SUB_CODE
SUBJECT
NAME
TOTAL
PAPER_NO
Marks_obt
MARKS
MARKS_ TOT
SUBJECT
RECEIVES
NAME
PARENT
PHONE _NO
GAURDS
STUDENT
SECTION
FEES
ROLL_NO
CLASS
NAME
1
N

RELATIONAL SCHEMA



CREATION OF TABLES

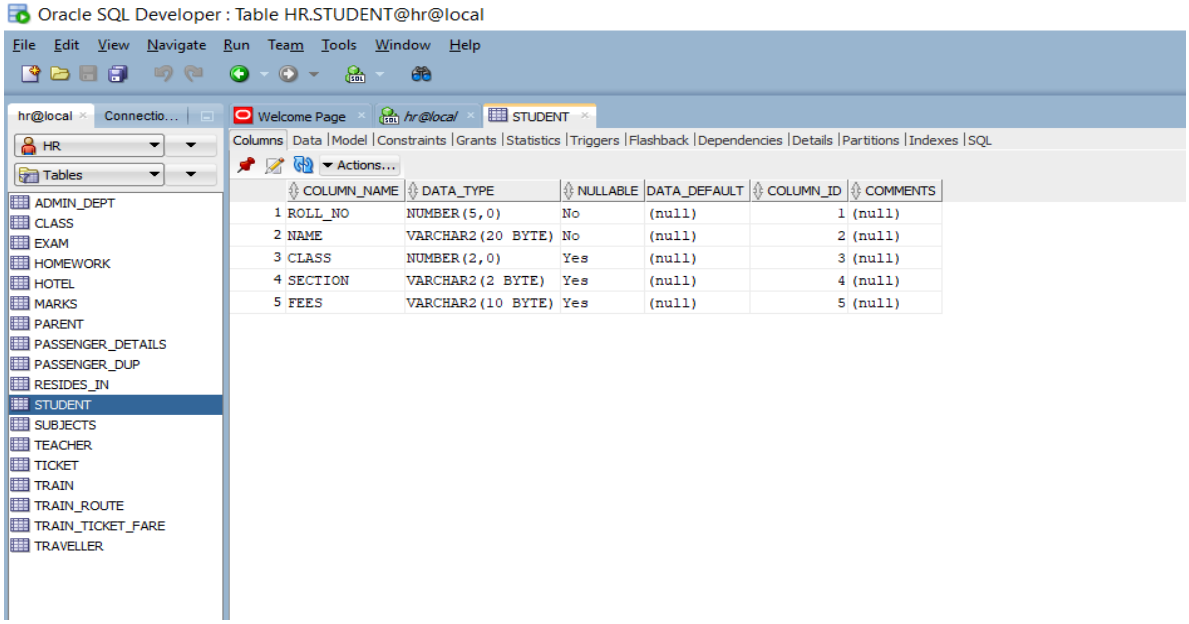
'Admin_dpt':



CODE:

```
create table Admin_Dept(
AD_ID number(5) constraint admin_pk primary key,
NAME varchar(20) not null,
Designation varchar(20) not null,
qualification varchar(10) not null,
email varchar(20) not null,
phone number(10) not null,
salary number(5));
```

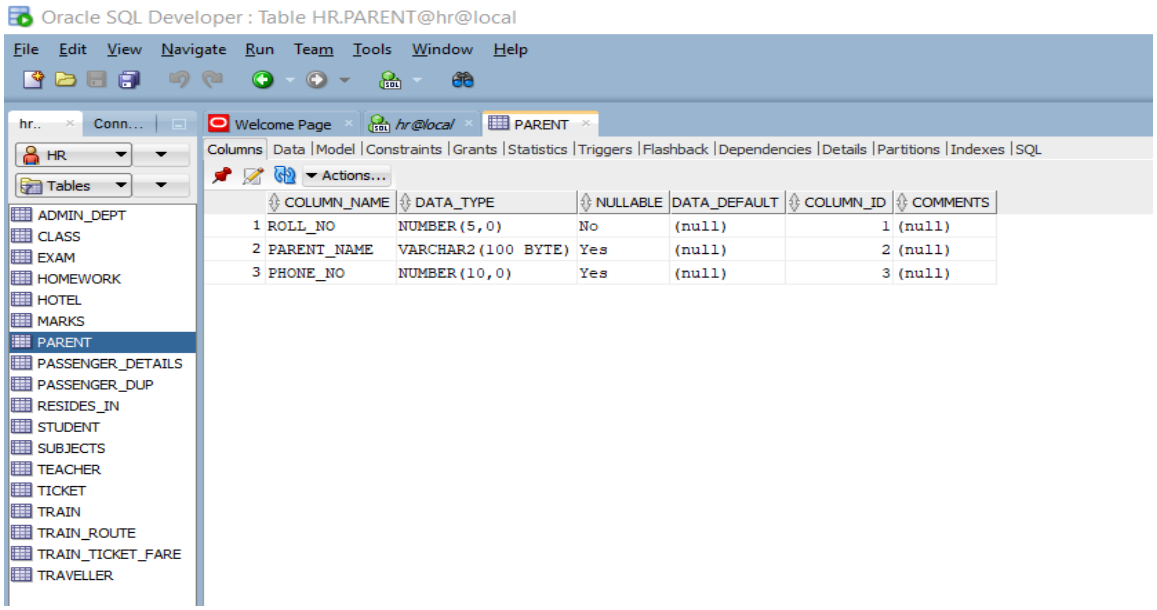
‘Student’:



CODE:

```
create table student(
roll_no number(5) constraint roll_pk primary key,
name varchar(20) constraint n_null not null,
class number(2),
section varchar(2), fees varchar(10));
```

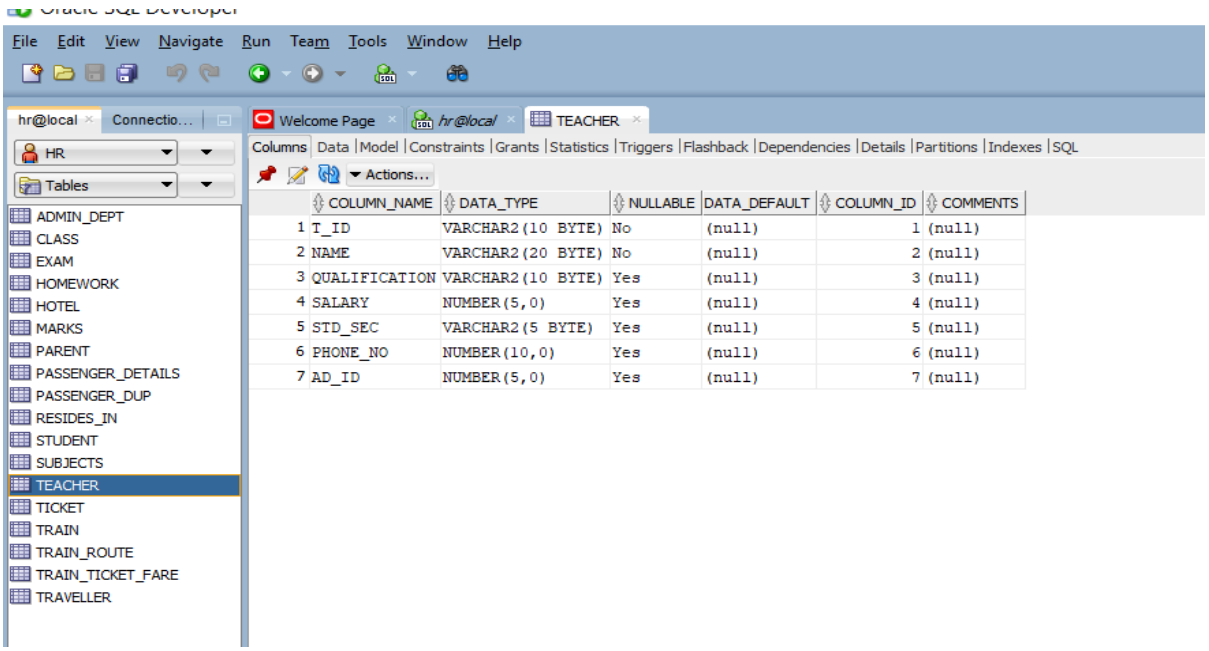
'Parent':



CODE:

```
Create table parent(roll_no number(5) references student,
Parent_name varchar(100),
Phone_No number(10),
primary key(Parent_name,roll_no) ON DELETE CASCADE);
```

'Teacher':

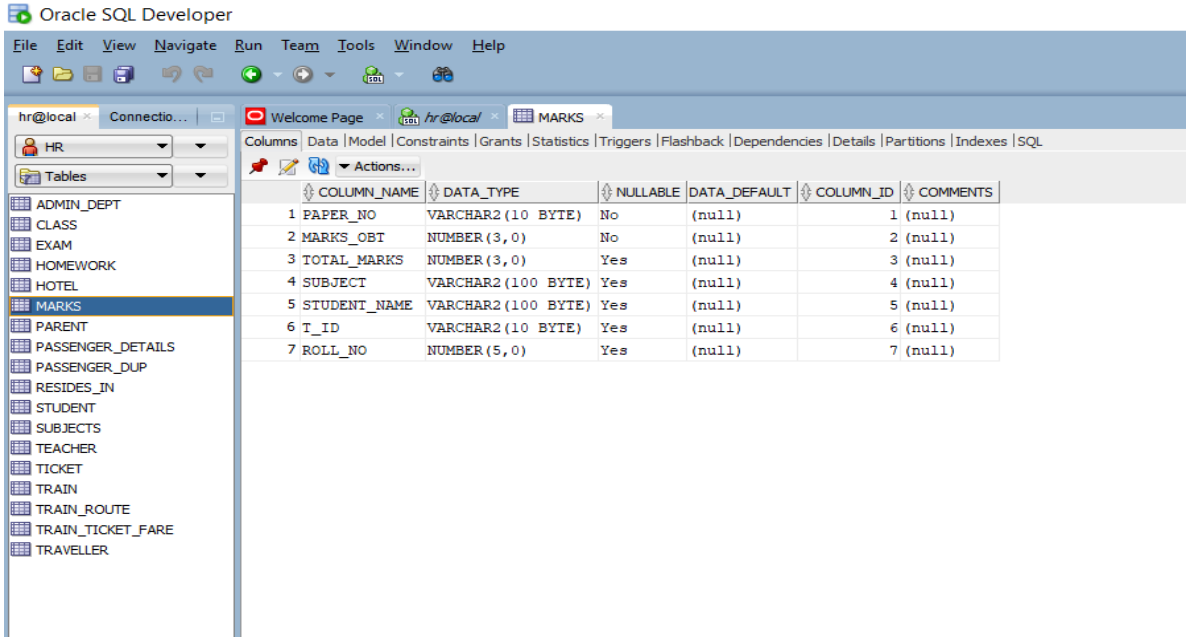


	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	T_ID	VARCHAR2(10 BYTE)	No	(null)	1	(null)
2	NAME	VARCHAR2(20 BYTE)	No	(null)	2	(null)
3	QUALIFICATION	VARCHAR2(10 BYTE)	Yes	(null)	3	(null)
4	SALARY	NUMBER(5,0)	Yes	(null)	4	(null)
5	STD_SEC	VARCHAR2(5 BYTE)	Yes	(null)	5	(null)
6	PHONE_NO	NUMBER(10,0)	Yes	(null)	6	(null)
7	AD_ID	NUMBER(5,0)	Yes	(null)	7	(null)

CODE:

```
create table Teacher(
t_id varchar(10) constraint t_id_pk primary key,
name varchar(20) constraint name_nnull not null,
Qualification varchar(10),
salary number(5),
std_sec varchar(5),
phone_no number(10),
Ad_id number(5),
constraint teacher_adid_fk foreign key(AD_ID) references Admin_Dept(AD_ID));
```

'Marks':

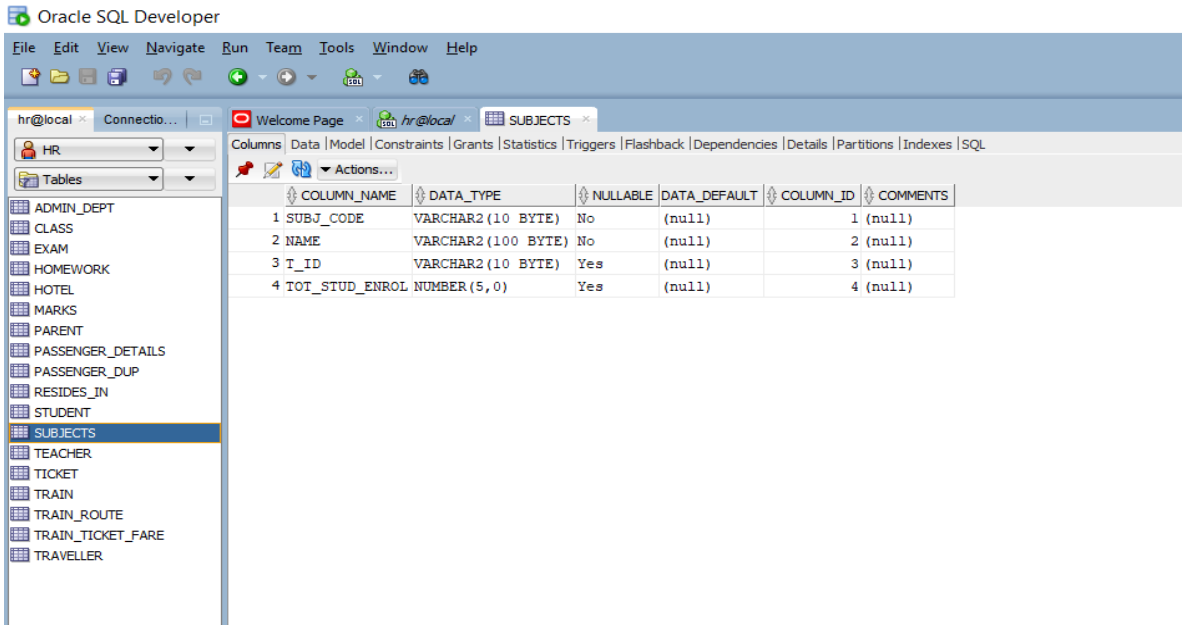


	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	PAPER_NO	VARCHAR2(10 BYTE)	No	(null)	1	(null)
2	MARKS_OBT	NUMBER(3,0)	No	(null)	2	(null)
3	TOTAL_MARKS	NUMBER(3,0)	Yes	(null)	3	(null)
4	SUBJECT	VARCHAR2(100 BYTE)	Yes	(null)	4	(null)
5	STUDENT_NAME	VARCHAR2(100 BYTE)	Yes	(null)	5	(null)
6	T_ID	VARCHAR2(10 BYTE)	Yes	(null)	6	(null)
7	ROLL_NO	NUMBER(5,0)	Yes	(null)	7	(null)

CODE:

```
Create table marks(
Paper_No  varchar(10)  constraint mark_pk primary key,
Marks_obt  number(3)  constraint mk_obt_nnull not null,
Total_marks number(3),
Subject varchar(100),
Student_name varchar(100),
T_ID varchar(10),
Roll_No number(5),
Constraint mk_fk1 foreign key(T_ID) references teacher(T_ID),
Constraint mk_fk2 foreign key(Roll_No) references student(Roll_No));
```

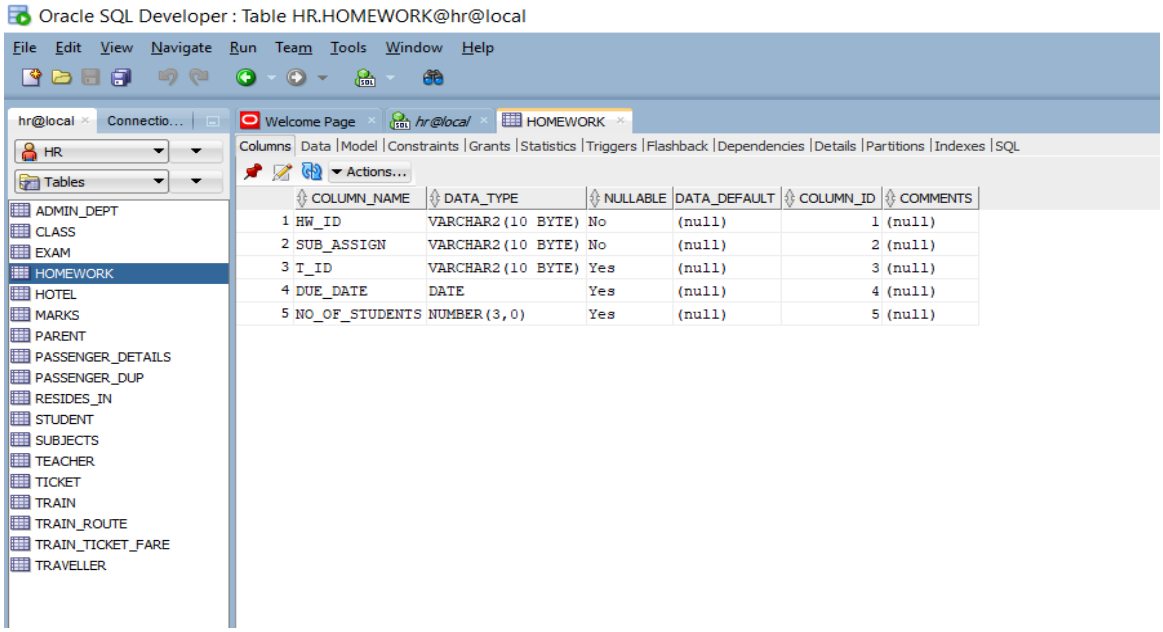
‘Subjects’:



CODE:

```
Create table subjects(
Subj_code varchar(10) constraint sub_code_pk primary key,
Name varchar(100) constraint subn_nnull not null,
T_ID varchar(10),
Tot_stud_enrol number(5),
Constraint subt_fk foreign key(T_ID) references teacher(T_ID));
```

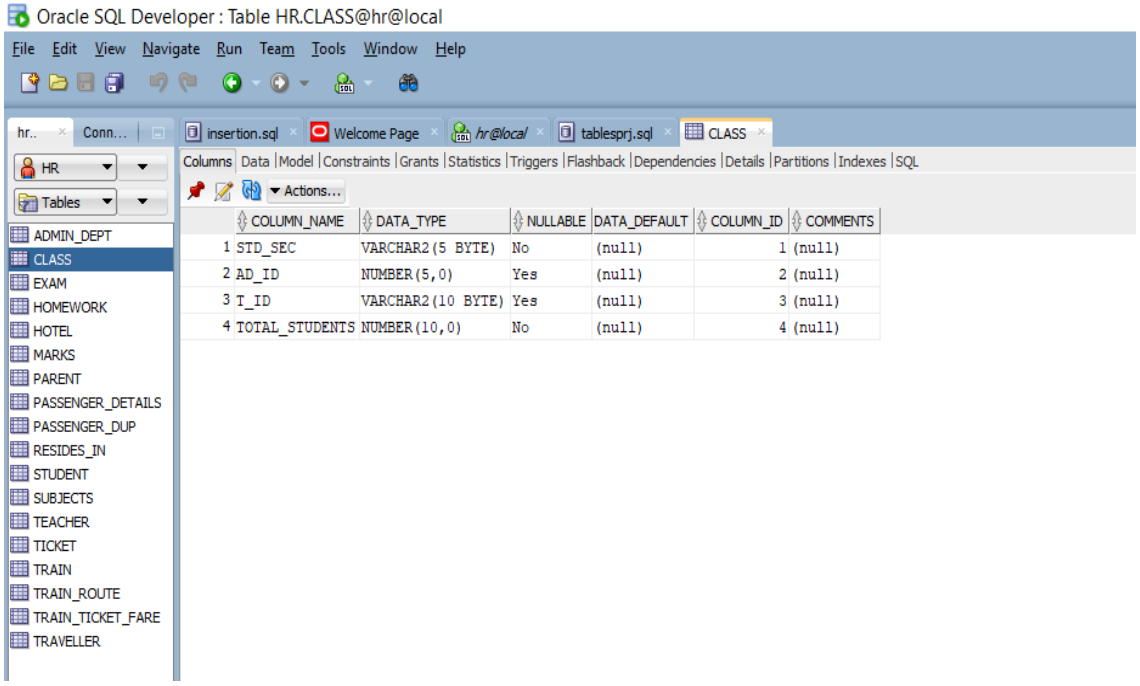
'Homework':



CODE:

```
Create table Homework(
Hw_id varchar(10) constraint hw_id_pk primary key,
Sub_assign varchar(10) constraint sub_nnull not null,
T_ID varchar(10),
Due_date date,
No_of_students number(3),
Constraint hw_fk foreign key(T_ID) references teacher(T_ID));
```

‘Class’:

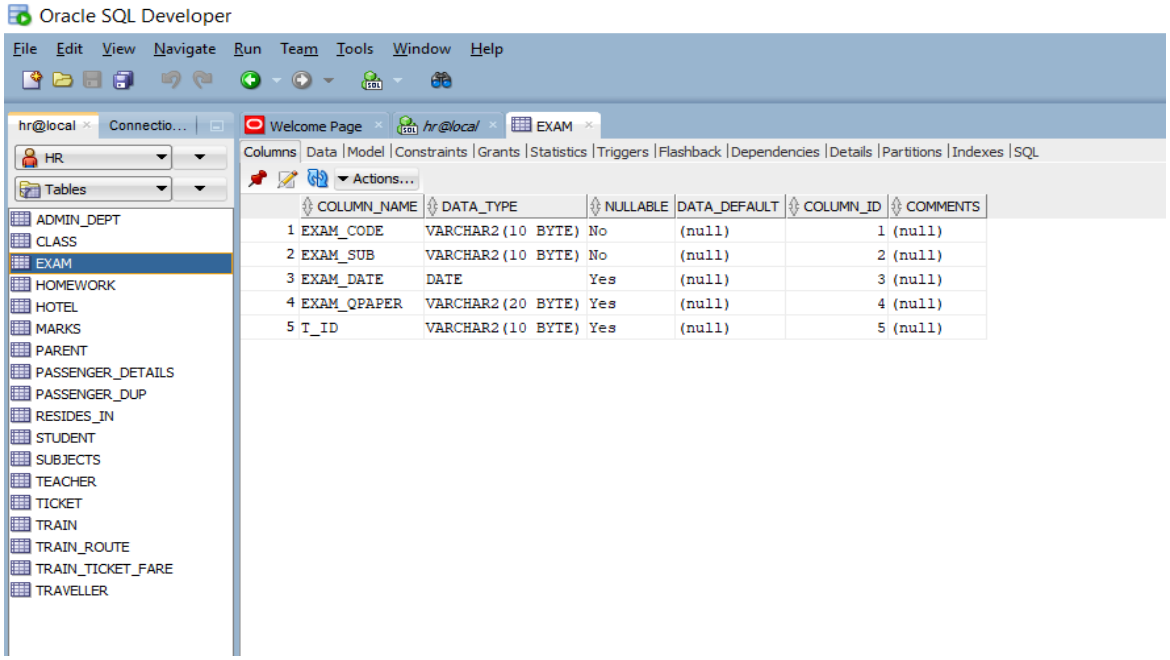


	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	STD_SEC	VARCHAR2(5 BYTE)	No	(null)	1	(null)
2	AD_ID	NUMBER(5,0)	Yes	(null)	2	(null)
3	T_ID	VARCHAR2(10 BYTE)	Yes	(null)	3	(null)
4	TOTAL_STUDENTS	NUMBER(10,0)	No	(null)	4	(null)

CODE:

```
Create table Class(
Std_sec varchar(5) constraint std_sc_pk primary key,
AD_ID number(5),
t_id varchar(10),
Total_students number(10) constraint ts_nnull not null,
Constraint cl_ad_fk foreign key(AD_ID) references Admin_Dept(AD_ID),
CONSTRAINT C_T_FK foreign key(t_id) references Teacher(t_id));
```

‘Exam’:



	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	EXAM_CODE	VARCHAR2(10 BYTE)	No	(null)	1	(null)
2	EXAM_SUB	VARCHAR2(10 BYTE)	No	(null)	2	(null)
3	EXAM_DATE	DATE	Yes	(null)	3	(null)
4	EXAM_QPAPER	VARCHAR2(20 BYTE)	Yes	(null)	4	(null)
5	T_ID	VARCHAR2(10 BYTE)	Yes	(null)	5	(null)

CODE:

```
Create table Exam(
Exam_code varchar(10) constraint ex_cd_pk primary key,
Exam_sub varchar(10) constraint ex_sb_nnull not null,
Exam_date date,
Exam_qpaper varchar(20),
T_ID varchar(10),
Constraint ex_fk foreign key(T_ID) references teacher(T_ID));
```

INSERTION OF DATA:

'Admin_dept':

INSERT INTO ADMIN_DEPT values(00001,'Anirudh Karanam','PRINCIPAL','PhD Chem','anirudh@gmail.com',8688456460,20000);

INSERT INTO ADMIN_DEPT VALUES(00002,'Yeswin','HEADMASTER','PhD Hist','yeswin@gmail.com',9912342395,15000);

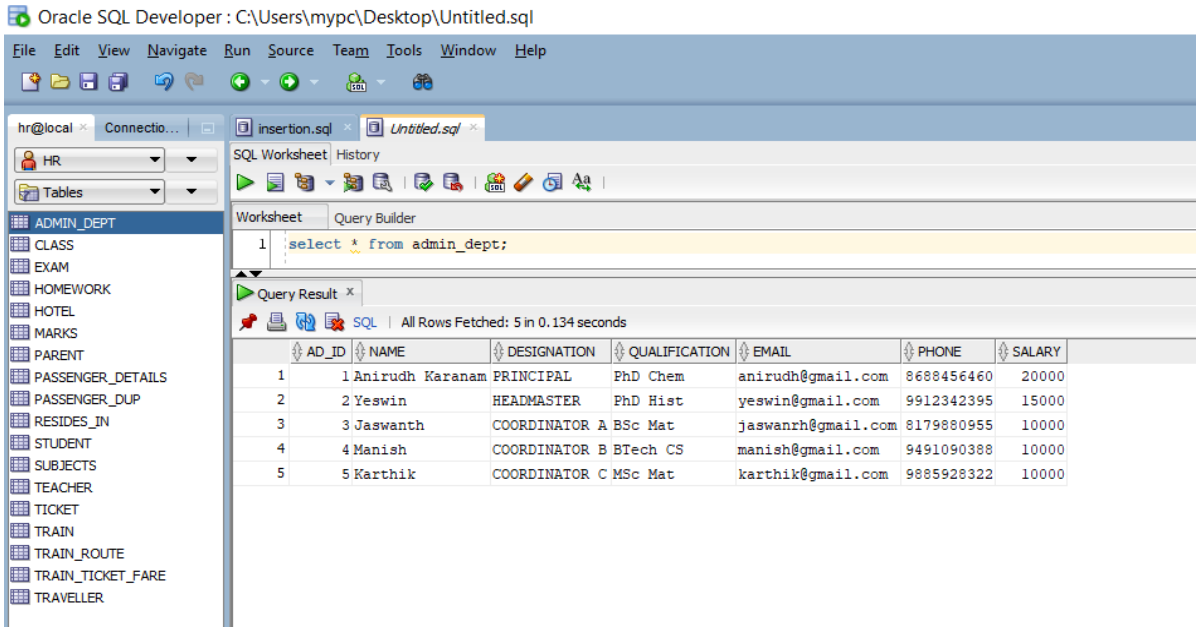
INSERT INTO ADMIN_DEPT VALUES(00003,'Jaswanth','COORDINATOR A','BSc Mat','jaswanrh@gmail.com',8179880955,10000);

INSERT INTO ADMIN_DEPT VALUES(00004,'Manish','COORDINATOR B','BTech CS','manish@gmail.com',9491090388,10000);

INSERT INTO ADMIN_DEPT VALUES(00005,'Karthik','COORDINATOR C','MSc Mat','karthik@gmail.com',9885928322,10000);

commit;

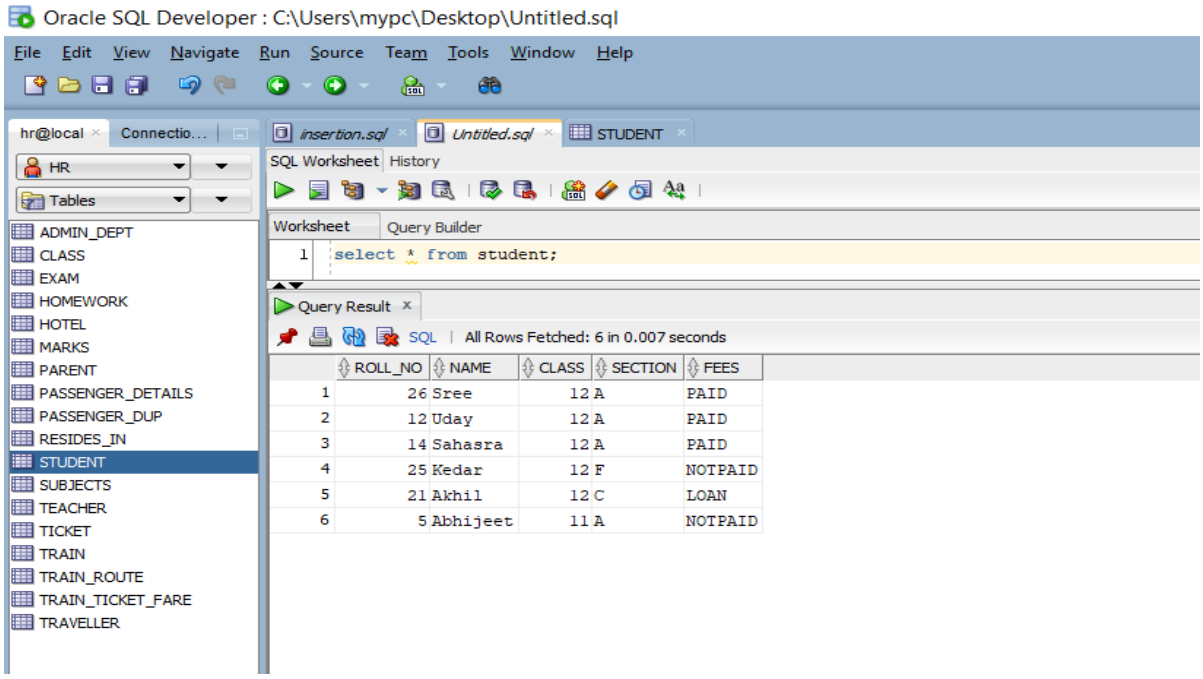
OUTPUT



'Student':

```
INSERT INTO STUDENT VALUES(26,'Sree',12,'A','PAID');
INSERT INTO STUDENT VALUES(12,'Uday',12,'A','PAID');
INSERT INTO STUDENT VALUES(14,'Sahasra',12,'A','PAID');
INSERT INTO STUDENT VALUES(25,'Kedar',12,'F','NOTPAID');
INSERT INTO STUDENT VALUES(21,'Akhil',12,'C','LOAN');
INSERT INTO STUDENT VALUES(05,'Abhijeet',11,'A','NOTPAID');
commit;
```

OUTPUT



'Parent':

INSERT INTO PARENT VALUES(14,'BadriNath',9866006460);

INSERT INTO PARENT VALUES(12,'Lakshmi',9898090990);

INSERT INTO PARENT VALUES(21,'Raghuram',8989898989);

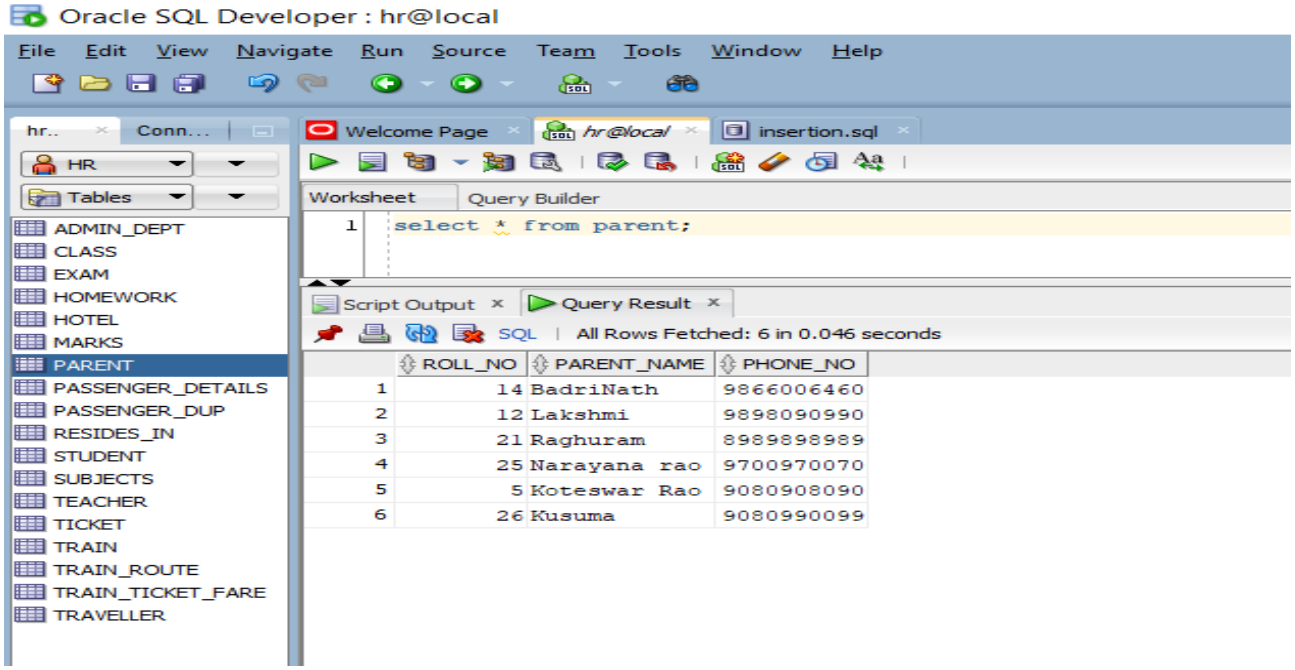
INSERT INTO PARENT VALUES(25,'Narayana rao',9700970070);

INSERT INTO PARENT VALUES(05,'Koteswar Rao',9080908090);

INSERT INTO PARENT VALUES(26,'Kusuma',9080990099);

commit;

OUTPUT



'Teacher':

INSERT INTO TEACHER VALUES('PMT001','Roy Santiago','BSc',4000,'CL001',6767676767,3);

INSERT INTO TEACHER VALUES('TMP002','Dinesh Kumar','BA',3500,'CL018',8989898989,4);

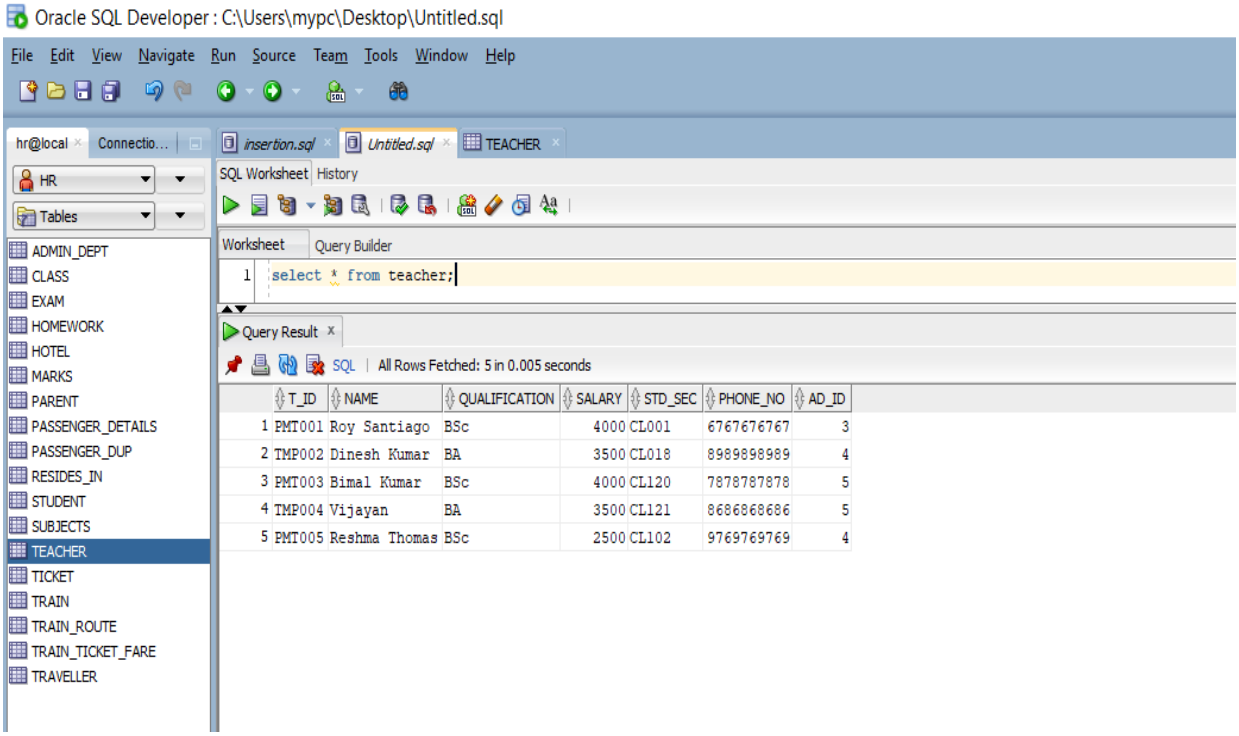
INSERT INTO TEACHER VALUES('PMT003','Bimal Kumar','BSc',4000,'CL120',7878787878,5);

INSERT INTO TEACHER VALUES('TMP004','Vijayan','BA',3500,'CL121',8686868686,5);

INSERT INTO TEACHER VALUES('PMT005','Reshma Thomas','BSc',2500,'CL102',9769769769,4);

commit;

OUTPUT



'Homework':

INSERT INTO HOMEWORK VALUES('HW1001','PHY','PMT003',DATE'2020-06-08',33);

INSERT INTO HOMEWORK VALUES('HW1002','CHEM','PMT005',DATE'2020-06-06',29);

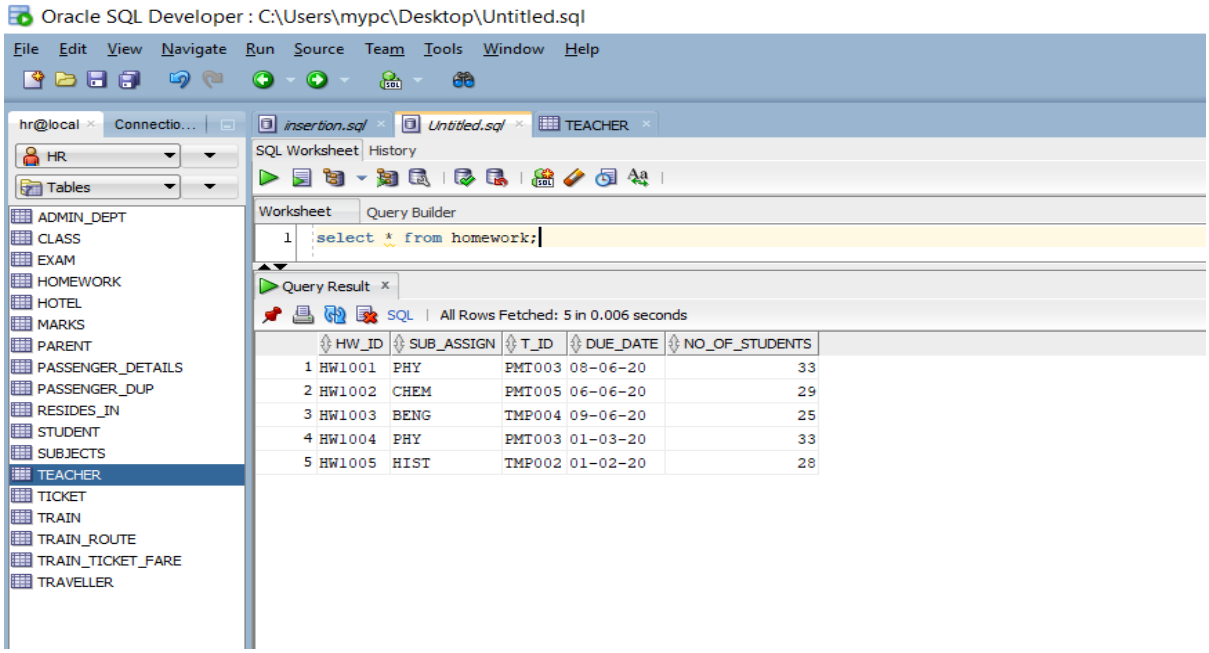
INSERT INTO HOMEWORK VALUES('HW1003','BENG','TMP004',DATE'2020-06-09',25);

INSERT INTO HOMEWORK VALUES('HW1004','PHY','PMT003',DATE'2020-03-01',33);

INSERT INTO HOMEWORK VALUES('HW1005','HIST','TMP002',DATE'2020-02-01',28);

commit;

OUTPUT



‘Exam’

INSERT INTO EXAM VALUES('EXM001','PHY',DATE'2020-06-08','qpaper1.docx','PMT001');

INSERT INTO EXAM VALUES('EXM002','HIST',DATE'2020-06-07','qpaper2.docx','TMP002');

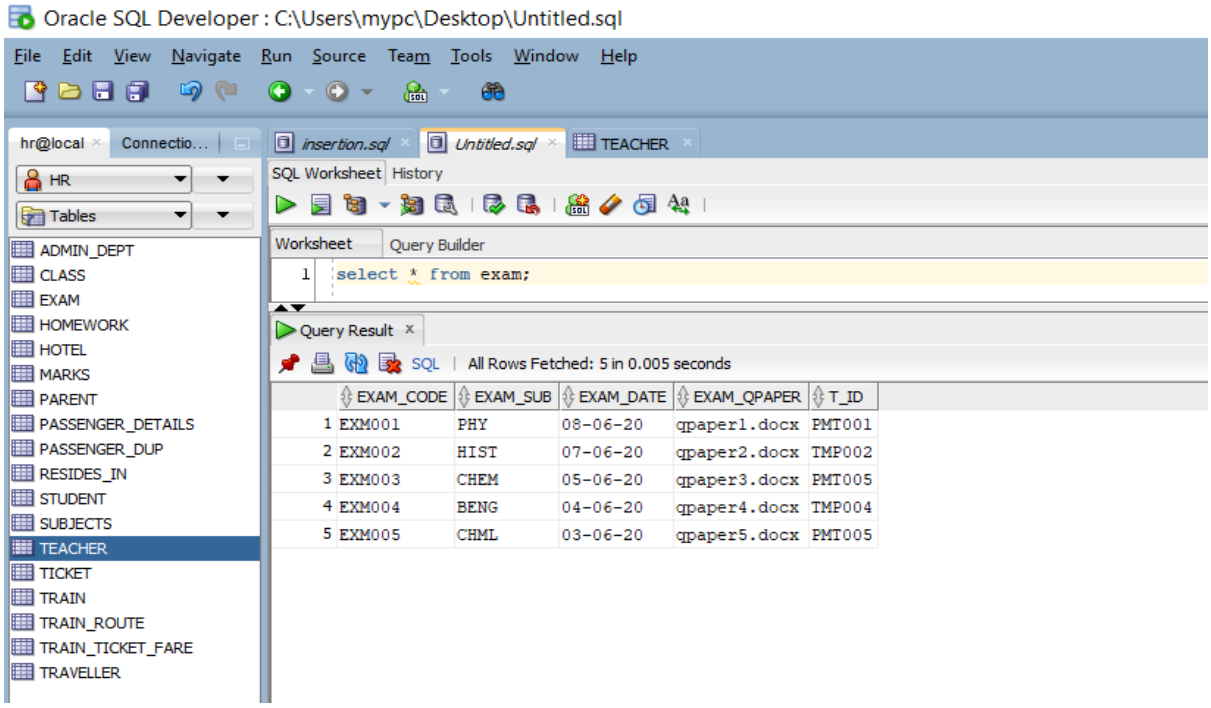
INSERT INTO EXAM VALUES('EXM003','CHEM',DATE'2020-06-05','qpaper3.docx','PMT005');

INSERT INTO EXAM VALUES('EXM004','BENG',DATE'2020-06-04','qpaper4.docx','TMP004');

INSERT INTO EXAM VALUES('EXM005','CHML',DATE'2020-06-03','qpaper5.docx','PMT005');

commit;

OUTPUT



‘Class’

INSERT INTO CLASS VALUES('12A',3,'PMT001',33);

INSERT INTO CLASS VALUES('12C',4,'TMP002',28);

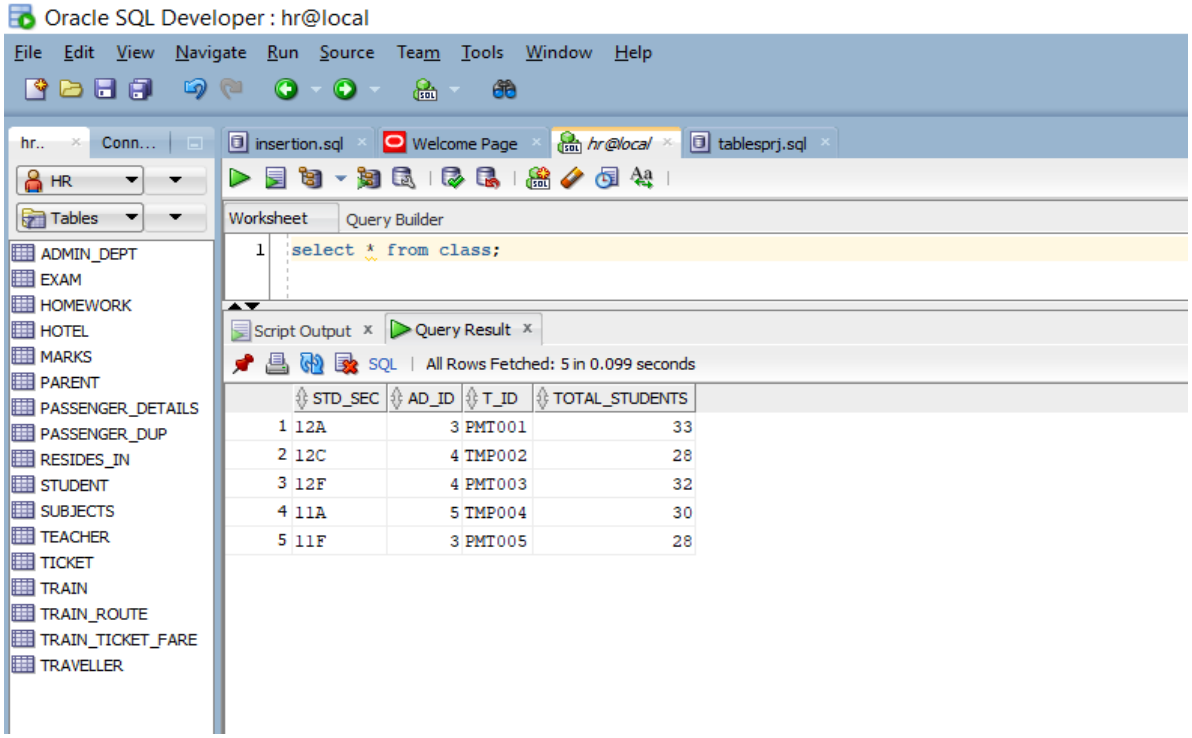
INSERT INTO CLASS VALUES('12F',4,'PMT003',32);

INSERT INTO CLASS VALUES('11A',5,'TMP004',30);

INSERT INTO CLASS VALUES('11F',3,'PMT005',28);

Commit;

OUTPUT



‘Subjects’

INSERT INTO SUBJECTS VALUES('PHY1001','Introduction to Physics','PMT003',70);

INSERT INTO SUBJECTS VALUES('CHY1001','Introduction to Chemistry','PMT001',35);

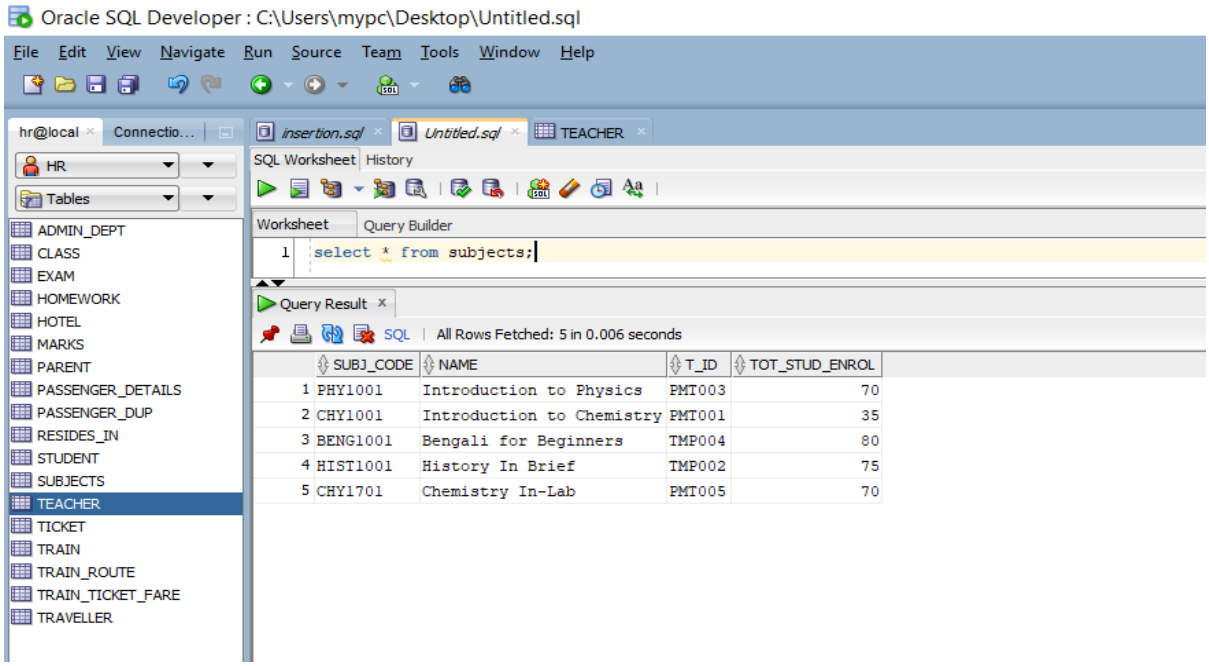
INSERT INTO SUBJECTS VALUES('BENG1001','Bengali for Beginners','TMP004',80);

INSERT INTO SUBJECTS VALUES('HIST1001','History In Brief','TMP002',75);

INSERT INTO SUBJECTS VALUES('CHY1701','Chemistry In-Lab','PMT005',70);

Commit;

OUTPUT



‘Marks’

INSERT INTO MARKS VALUES('SL001',98,100,'PHYSICS','Sahasra','PMT003',14);

INSERT INTO MARKS VALUES('SL0001',97,100,'PHYSICS','Sree','PMT003',26);

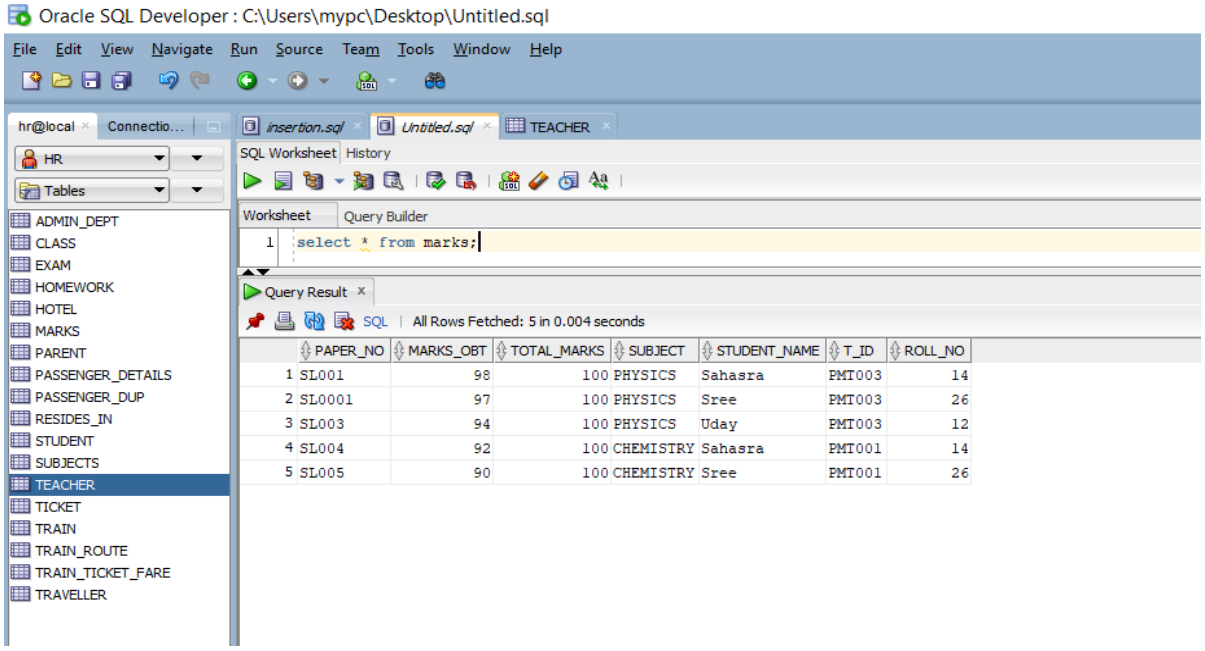
INSERT INTO MARKS VALUES('SL003',94,100,'PHYSICS','Uday','PMT003',12);

INSERT INTO MARKS VALUES('SL004',92,100,'CHEMISTRY','Sahasra','PMT001',14);

INSERT INTO MARKS VALUES('SL005',90,100,'CHEMISTRY','Sree','PMT001',26);

commit;

OUTPUT



RETRIVAL OF DATA:

1.Students whose marks are greater than 95 (first division eligibility):

Worksheet | Query Builder

```
SELECT student_name,marks_obt FROM marks WHERE marks_obt>95;
```

Query Result

All Rows Fetched: 2 in 0.077 seconds

	STUDENT_NAME	MARKS_OBT
1	Sahasra	98
2	Sree	97

2.Faculty staff who has a BTech degree in Computer Science:

Worksheet | Query Builder

```
SELECT AD_ID,NAME FROM ADMIN_DEPT WHERE QUALIFICATION='BTech CS';
```

Query Result

All Rows Fetched: 1 in 0.002 seconds

	AD_ID	NAME
1	4	Manish

3.Students who haven't paid their fees:

Worksheet | Query Builder

```
SELECT NAME,ROLL_NO,FEES FROM STUDENT WHERE FEES='NOTPAID';
```

Query Result

All Rows Fetched: 2 in 0.027 seconds

	NAME	ROLL_NO	FEES
1	Kedar	25	NOTPAID
2	Abhijeet	5	NOTPAID

4.Students who have timely paid their fees:

```
SELECT NAME,ROLL_NO,FEES FROM STUDENT WHERE FEES='PAID';
```

Query Result

SQL | All Rows Fetched: 3 in 0.004 seconds

	NAME	ROLL_NO	FEES
1	Sree	26	PAID
2	Uday	12	PAID
3	Sahasra	14	PAID

5.Subjects where more than 40 students have enrolled:

Worksheet | Query Builder

```
SELECT NAME,SUBJ_CODE FROM SUBJECTS WHERE TOT_STUD_ENROL>40;
```

Query Result

SQL | All Rows Fetched: 4 in 0.029 seconds

	NAME	SUBJ_CODE
1	Introduction to Physics	PHY1001
2	Bengali for Beginners	BENG1001
3	History In Brief	HIST1001
4	Chemistry In-Lab	CHY1701

6.Finding Teachers who are permanent in the school:

Worksheet | Query Builder

```
SELECT NAME,T_ID FROM TEACHER WHERE T_ID LIKE'P%';
```

Query Result

SQL | All Rows Fetched: 3 in 0.073 seconds

	NAME	T_ID
1	Roy Santiago	PMT001
2	Bimal Kumar	PMT003
3	Reshma Thomas	PMT005

7.Finding teachers supervised by a certain coordinator:

Worksheet | Query Builder

```
1  SELECT NAME, T_ID FROM TEACHER WHERE AD_ID=5;
```

Query Result

SQL | All Rows Fetched: 2 in 0.051 seconds

	NAME	T_ID
1	Bimal Kumar	PMT003
2	Vijayan	TMP004

8.Faculty who have BSc degree:

Worksheet | Query Builder

```
1  SELECT NAME,AD_ID FROM ADMIN_DEPT WHERE QUALIFICATION LIKE'BSc%';
```

Query Result

SQL | All Rows Fetched: 1 in 0.064 seconds

	NAME	AD_ID
1	Jaswanth	3

MODIFICATION OF DATA:

1.Updation of Parent's phone number due to unforeseen circumstances.

Worksheet | Query Builder

```
UPDATE PARENT  SET phone_no = 9912342395 WHERE roll_no = 14;
select * from PARENT;
```

Script Output | Query Result

All Rows Fetched: 6 in 0.004 seconds

	ROLL_NO	PARENT_NAME	PHONE_NO
1	14	BadriNath	9912342395
2	12	Lakshmi	9898090990
3	21	Raghuram	8989898989
4	25	Narayana rao	9700970070
5	5	Koteswar Rao	9080908090
6	26	Kusuma	9080990099

2.Updation of marks in an exam, due to internal error- by the teacher concerned only.

Worksheet | Query Builder

```
UPDATE MARKS set marks_obt = 91 WHERE paper_no = 'SL003';
SELECT * FROM MARKS;
```

Script Output | Query Result

All Rows Fetched: 5 in 0.029 seconds

	PAPER_NO	MARKS_OBT	TOTAL_MARKS	SUBJECT	STUDENT_NAME	T_ID	ROLL_NO
1	SL001	98	100	PHYSICS	Sahasra	PMT003	14
2	SL0001	97	100	PHYSICS	Sree	PMT003	26
3	SL003	91	100	PHYSICS	Uday	PMT003	12
4	SL004	92	100	CHEMISTRY	Sahasra	PMT001	14
5	SL005	90	100	CHEMISTRY	Sree	PMT001	26

3.Updation of homework deadline by the teacher for multiple reasons:

Worksheet | Query Builder

```
UPDATE HOMEWORK SET DUE_DATE=DATE'2020-06-10' WHERE HW_ID='HW1001';
SELECT * FROM homework;
```

Script Output | Query Result

All Rows Fetched: 5 in 0.006 seconds

	HW_ID	SUB_ASSIGN	T_ID	DUE_DATE	NO_OF_STUDENTS
1	HW1001	PHY	PMT003	10-06-20	33
2	HW1002	CHEM	PMT005	06-06-20	29
3	HW1003	BENG	TMP004	09-06-20	25
4	HW1004	PHY	PMT003	01-03-20	33
5	HW1005	HIST	TMP002	01-02-20	28

4.Updation of exam paper after moderation:

Worksheet | Query Builder

```
UPDATE EXAM SET EXAM_QPAPER='qpaperx.docx' WHERE EXAM_CODE='EXM004';
SELECT * FROM exam;
```

Script Output | Query Result

All Rows Fetched: 5 in 0.022 seconds

	EXAM_CODE	EXAM_SUB	EXAM_DATE	EXAM_QPAPER	T_ID
1	EXM001	PHY	08-06-20	qpaper1.docx	PMT001
2	EXM002	HIST	07-06-20	qpaper2.docx	TMP002
3	EXM003	CHEM	05-06-20	qpaper3.docx	PMT005
4	EXM004	BENG	04-06-20	qpaperx.docx	TMP004
5	EXM005	CHML	03-06-20	qpaper5.docx	PMT005

DELETION OF DATA:

1.Deletion of student's data if he/she is no longer part of the school:

BEFORE:

Worksheet | Query Builder

```
select * from student;
```

Query Result

SQL | All Rows Fetched: 6 in 0.007 seconds

	ROLL_NO	NAME	CLASS	SECTION	FEES
1	26	Sree	12	A	PAID
2	12	Uday	12	A	PAID
3	14	Sahasra	12	A	PAID
4	25	Kedar	12	F	NOTPAID
5	21	Akhil	12	C	LOAN
6	5	Abhijeet	11	A	NOTPAID

AFTER:

Worksheet | Query Builder

```
DELETE FROM STUDENT WHERE ROLL_NO=25;
SELECT * FROM STUDENT;
```

Script Output | Query Result

SQL | All Rows Fetched: 5 in 0.003 seconds

	ROLL_NO	NAME	CLASS	SECTION	FEES
1	26	Sree	12	A	PAID
2	12	Uday	12	A	PAID
3	14	Sahasra	12	A	PAID
4	21	Akhil	12	C	LOAN
5	5	Abhijeet	11	A	NOTPAID

2.Deletion of homework due to some reasons faculty may decides

BEFORE:

Worksheet | Query Builder

```
select * from homework;
```

Script Output | Query Result

SQL | All Rows Fetched: 5 in 0.043 seconds

	HW_ID	SUB_ASSIGN	T_ID	DUE_DATE	NO_OF_STUDENTS
1	HW1001	PHY	PMT003	10-06-20	33
2	HW1002	CHEM	PMT005	06-06-20	29
3	HW1003	BENG	TMP004	09-06-20	25
4	HW1004	PHY	PMT003	01-03-20	33
5	HW1005	HIST	TMP002	01-02-20	28

AFTER:

Worksheet | Query Builder

```
DELETE FROM HOMEWORK where hw_id='HW1002';
select * from homework;
```

Script Output | Query Result

SQL | All Rows Fetched: 4 in 0.002 seconds

	HW_ID	SUB_ASSIGN	T_ID	DUE_DATE	NO_OF_STUDENTS
1	HW1001	PHY	PMT003	10-06-20	33
2	HW1003	BENG	TMP004	09-06-20	25
3	HW1004	PHY	PMT003	01-03-20	33
4	HW1005	HIST	TMP002	01-02-20	28

3.Deletion of a subject from a list- if it is no longer offered by the school

BEFORE:

Worksheet | Query Builder

```
select * from subjects;
```

Script Output | Query Result

SQL | All Rows Fetched: 5 in 0.002 seconds

	SUBJ_CODE	NAME	T_ID	TOT_STUD_ENROL
1	CHY1001	Introduction to Chemistry	PMT001	35
2	BENG1001	Bengali for Beginners	TMP004	80
3	HIST1001	History In Brief	TMP002	75
4	CHY1701	Chemistry In-Lab	PMT005	70
5	PHY1001	Introduction to Physics	PMT003	70

AFTER:

Worksheet | Query Builder

```
DELETE FROM SUBJECTS WHERE SUBJ_CODE='CHY1701';
select * from subjects;
```

Script Output | Query Result

SQL | All Rows Fetched: 4 in 0.004 seconds

	SUBJ_CODE	NAME	T_ID	TOT_STUD_ENROL
1	CHY1001	Introduction to Chemistry	PMT001	35
2	BENG1001	Bengali for Beginners	TMP004	80
3	HIST1001	History In Brief	TMP002	75
4	PHY1001	Introduction to Physics	PMT003	70

4.Deletion of a of exam in very critical cases where there is leakage of paper and extreme malpractises.

BEFORE:

Worksheet | Query Builder

```
1  select * from exam;
```

Script Output | Query Result

SQL | All Rows Fetched: 5 in 0.042 seconds

	EXAM_CODE	EXAM_SUB	EXAM_DATE	EXAM_QPAPER	T_ID
1	EXM001	PHY	08-06-20	qpaper1.docx	PMT001
2	EXM002	HIST	07-06-20	qpaper2.docx	TMP002
3	EXM003	CHEM	05-06-20	qpaper3.docx	PMT005
4	EXM004	BENG	04-06-20	qpaperx.docx	TMP004
5	EXM005	CHML	03-06-20	qpaper5.docx	PMT005

AFTER:

Worksheet | Query Builder

```
1  DELETE FROM exam where exam_code='EXM002';
2  select * from exam;
```

Script Output | Query Result

SQL | All Rows Fetched: 4 in 0.003 seconds

	EXAM_CODE	EXAM_SUB	EXAM_DATE	EXAM_QPAPER	T_ID
1	EXM001	PHY	08-06-20	qpaper1.docx	PMT001
2	EXM003	CHEM	05-06-20	qpaper3.docx	PMT005
3	EXM004	BENG	04-06-20	qpaperx.docx	TMP004
4	EXM005	CHML	03-06-20	qpaper5.docx	PMT005

USAGE OF NVL AND NULLIF FUNCTIONS

i)nullif function:

select nullif(marks_obt,0)"Marks",student_name,roll_no from marks;

Worksheet | Query Builder

```
1 select marks_obt,student_name,roll_no from marks;
2 select nullif(marks_obt,0)"Marks",student_name,roll_no from marks;
```

Script Output | Query Result

SQL | All Rows Fetched: 8 in 0.003 seconds

	MARKS_OBT	STUDENT_NAME	ROLL_NO
1	98	Sahasra	14
2	97	Sree	26
3	91	Uday	12
4	92	Sahasra	14
5	90	Sree	26
6	0	Akhil	21
7	0	Abhijeet	5
8	0	Akhil	21

Worksheet | Query Builder

```
1 select marks_obt,student_name,roll_no from marks;
2 select nullif(marks_obt,0)"Marks",student_name,roll_no from marks;
```

Script Output | Query Result

SQL | All Rows Fetched: 8 in 0.001 seconds

	Marks	STUDENT_NAME	ROLL_NO
1	98	Sahasra	14
2	97	Sree	26
3	91	Uday	12
4	92	Sahasra	14
5	90	Sree	26
6	(null)	Akhil	21
7	(null)	Abhijeet	5
8	(null)	Akhil	21

ii)nvl function:

SELECT name,roll_no,class,nvl(fees,'NOT REGISTERED') from STUDENT;

Script Output | Query Result

SQL | All Rows Fetched: 8 in 0.002 seconds

	NAME	ROLL_NO	CLASS	FEES
1	Sree	26	12	PAID
2	Uday	12	12	PAID
3	Sahasra	14	12	PAID
4	Anirudh	54	11	(null)
5	Akhil	21	12	LOAN
6	Abhijeet	5	11	NOTPAID
7	Yeswin	48	11	(null)
8	Jaswanth	36	11	(null)

Script Output | Query Result

SQL | All Rows Fetched: 8 in 0.003 seconds

	NAME	ROLL_NO	CLASS	NVL(FEES,'NOTREGISTERED')
1	Sree	26	12	PAID
2	Uday	12	12	PAID
3	Sahasra	14	12	PAID
4	Anirudh	54	11	NOT REGISTERED
5	Akhil	21	12	LOAN
6	Abhijeet	5	11	NOTPAID
7	Yeswin	48	11	NOT REGISTERED
8	Jaswanth	36	11	NOT REGISTERED

JOIN QUERY ORDERBY CLAUSE:

selectteacher.name,teacher.qualification,admin_dept.name,admin_dept.designation,admin_dept.ad_id from

admin_dept join teacher on admin_dept.ad_id = teacher.ad_id

order by admin_dept.ad_id asc;

Script Output | Query Result

SQL | All Rows Fetched: 5 in 0.003 seconds

	NAME	QUALIFICATION	NAME_1	DESIGNATION	AD_ID
1	Roy Santiago	BSc	Jaswanth	COORDINATOR A	3
2	Dinesh Kumar	BA	Manish	COORDINATOR B	4
3	Reshma Thomas	BSc	Manish	COORDINATOR B	4
4	Vijayan	BA	Karthik	COORDINATOR C	5
5	Bimal Kumar	BSc	Karthik	COORDINATOR C	5

UNCORELATED NESTED QUERY

select name from TEACHER

where t_id in

(select T_id from EXAM where Exam_date between to_date('02-06-2020','dd-mm-yyyy')and to_date('07-06-2020','dd-mm-yyyy'));

Worksheet | Query Builder

```
1 select name from TEACHER
2 where t_id in
3 (select T_id from EXAM where Exam_date between to_date('02-06-2020','dd-mm-yyyy')and to_date('07-06-2020','dd-mm-yyyy'));
```

Script Output | Query Result

SQL | All Rows Fetched: 2 in 0.003 seconds

	NAME
1	Reshma Thomas
2	Vijayan

CORRELATED NESTED QUERY

SELECT * FROM teacher where exists(select t_id from subjects where teacher.t_id = subjects.t_id);

Worksheet | Query Builder

```
1 SELECT * FROM teacher where exists(select t_id from subjects where teacher.t_id = subjects.t_id);
```

Script Output | Query Result

SQL | All Rows Fetched: 4 in 0.002 seconds

	T_ID	NAME	QUALIFICATION	SALARY	STD_SEC	PHONE_NO	AD_ID
1	PMT001	Roy Santiago	BSc	4000	CL001	6767676767	3
2	PMT003	Bimal Kumar	BSc	4000	CL120	7878787878	5
3	TMP002	Dinesh Kumar	BA	3500	CL018	8989898989	4
4	TMP004	Vijayan	BA	3500	CL121	8686868686	5

SET OPERATORS(MINUS)

select t_id from homework minus select t_id from exam;

Worksheet | Query Builder

```
1 select t_id from homework minus select t_id from exam;
```

Script Output | Query Result

SQL | All Rows Fetched: 2 in 0.001 seconds

	T_ID
1	PMT003
2	TMP002

QUERY GROUPBY HAVING WHERE CLAUSE

select name,count(t_id) from subjects GROUP BY name having count(t_id)=1;

Welcome Page | hr@local

Worksheet | Query Builder

```
1  select name,count(t_id) from subjects GROUP BY name having count(t_id)=1;
```

Query Result

SQL | All Rows Fetched: 5 in 0.077 seconds

	NAME	COUNT(T_ID)
1	Introduction to Physics	1
2	Bengali for Beginners	1
3	Instrumental Physics	1
4	Introduction to Chemistry	1
5	History In Brief	1

ONE QUERY INVOLVING OUTER JOIN

select teacher.name,teacher.qualification,teacher.phone_no,subjects.name,subjects.tot_stud_enrol

from teacher full outer join subjects on teacher.t_id = subjects.t_id;

Worksheet | Query Builder

```
1  select teacher.name,teacher.qualification,teacher.phone_no,subjects.name,subjects.tot_stud_enrol
2  from teacher full outer join subjects on teacher.t_id = subjects.t_id;
3
```

Script Output | Query Result

SQL | All Rows Fetched: 6 in 0.02 seconds

	NAME	QUALIFICATION	PHONE_NO	NAME_1	TOT_STUD_ENROL
1	Roy Santiago	BSc	6767676767	Introduction to Chemistry	35
2	Dinesh Kumar	BA	8989898989	History In Brief	75
3	Bimal Kumar	BSc	7878787878	Instrumental Physics	65
4	Bimal Kumar	BSc	7878787878	Introduction to Physics	70
5	Vijayan	BA	8686868686	Bengali for Beginners	80
6	Reshma Thomas	BSc	9769769769	(null)	(null)

PL/SQL STATEMENTS

1.RETRIVE THE FEE STATUS OF STUDENTS

```
set SERVEROUTPUT ON;
create or replace procedure fee_details is
cursor ta is select roll_no,name,fees from student;
s_roll number(4);
s_name varchar(15);
s_fee varchar(10);
begin
open ta;
loop
fetch ta into s_roll,s_name,s_fee;
exit when ta%notfound;
dbms_output.put_line(s_roll || ' ' || s_name ||' ' || s_fee);
end loop;
close ta;
end;
exec fee_details;
```

Worksheet | Query Builder

```
1  set SERVEROUTPUT ON;
2
3  create or replace procedure fee_details is
4  cursor ta is select roll_no,name,fees from student;
5  s_roll number(4);
6  s_name varchar(15);
7  s_fee varchar(10);
8  begin
9  open ta;
10 loop
11 fetch ta into s_roll,s_name,s_fee;
12 exit when ta%notfound;
13 dbms_output.put_line(s_roll || ' ' || s_name ||' ' || s_fee);
14 end loop;
15 close ta;
16 end;
17
18 exec fee_details;
19
20
21
22
```

Script Output | Task completed in 0.035 seconds

```
Procedure FEE_DETAILS compiled

26 Sree PAID
12 Uday PAID
14 Sahasra PAID
54 Anirudh
21 Akhil LOAN
5 Abhijeet NOTPAID
48 Yeswin
36 Jaswanth


PL/SQL procedure successfully completed.
```

2.DISPLAY TEACHER NAME OF CERTAIN SUBJECT

```
create or replace procedure t_name(Sub_code varchar) is
cursor tn is select name from TEACHER where t_id in(select T_ID from subjects where Subj_code=sub_code);
teacher_name TEACHER.name%type;
begin
for t_rec in tn loop
teacher_name:=t_rec.name;
dbms_output.put_line('teacher name is :' ||teacher_name);
end loop;
end;
/
exec t_name('CHY1001');
```

Worksheet | Query Builder

```
18
19
20 create or replace procedure t_name(Sub_code varchar) is
21 cursor tn is select name from TEACHER where t_id in(select T_ID from subjects where Subj_code=sub_code);
22 teacher_name TEACHER.name%type;
23 begin
24 for t_rec in tn loop
25 teacher_name:=t_rec.name;
26 dbms_output.put_line('teacher name is :' ||teacher_name);
27 end loop;
28 end;
29 /
30
31 exec t_name('CHY1001');
32
33
```

Script Output | Query Result

Task completed in 0.49 seconds

```
Procedure T_NAME compiled

teacher name is :Roy Santiago


PL/SQL procedure successfully completed.
```

3.DISPLAY NUMBER OF STUDENTS WHO PAID THE FEE

```
create or replace function totalSt
return integer
as
total integer:=0;
begin
select count(roll_no) into total from student where fees='PAID';
return total;
end totalSt;

declare
answer integer;
begin
answer:=totalst();
   dbms_output.put_line('total number of students paid is  ' || answer);
end;
```

Worksheet | Query Builder

```
32
33
34 create or replace function totalSt
35 return integer
36 as
37 total integer:=0;
38
39 begin
40 select count(roll_no) into total from student where fees='PAID';
41 return total;
42 end totalSt;
43
44 declare
45 answer integer;
46
47 begin
48 answer:=totalst();
49    dbms_output.put_line('total number of students paid is  ' || answer);
50 end;
51
52
```

Script Output | Query Result

Task completed in 0.396 seconds

```
Function TOTALST compiled

total number of students paid is  3


PL/SQL procedure successfully completed.
```

4.DISPLAY TOTAL NUMBER OF STUDENTS IN A PERTICULAR SECTION

```
create or replace function num_student(student_sec varchar)
return number
as
cal number;
begin
select Total_students into cal from Class where  Std_sec=student_sec;
return cal;
end num_student;

declare
ans number;
begin
ans:=num_student('12A');
   dbms_output.put_line('no of studens in this class are  ' || ans);
end;
```

Worksheet | Query Builder

```
58 create or replace function num_student(student_sec varchar)
59
60 return number
61 as
62 cal number;
63
64 begin
65 select Total_students into cal from Class where  Std_sec=student_sec;
66 return cal;
67 end num_student;
68
69 declare
70 ans number;
71
72 begin
73 ans:=num_student('12A');
74    dbms_output.put_line('no of studens in this class are  ' || ans);
75 end;
76
77
78
```

Script Output × | Query Result ×

Task completed in 0.128 seconds

```
Function NUM_STUDENT compiled

no of studens in this class are  33

PL/SQL procedure successfully completed.
```

5.WHEN TEACHER ID IS CHANGED CORRESPONDING RELATED DATA IS TO BE UPDATED.

```
create or replace trigger trg_af_up_t
after update of t_id on TEACHER
referencing new as new old as old
for each row
begin
update Class set t_id=:new.t_id where t_id =:old.t_id;
update Exam set T_ID=:new.t_id where T_ID =:old.t_id;
update Homework set T_ID=:new.t_id where T_ID =:old.t_id;
update marks set T_ID=:new.t_id where T_ID =:old.t_id;
update subjects set T_ID=:new.t_id where T_ID =:old.t_id;
end;

update TEACHER set t_id='PMT013' where t_id='PMT001';
select * from exam;
select * from class;
select * from marks;
select * from subjects;
```

Worksheet | Query Builder

```
/*Triggers*/
create or replace trigger trg_af_up_t
after update of t_id on TEACHER
referencing new as new old as old
for each row
begin
update Class set t_id=:new.t_id where t_id =:old.t_id;
update Exam set T_ID=:new.t_id where T_ID =:old.t_id;
update Homework set T_ID=:new.t_id where T_ID =:old.t_id;
update marks set T_ID=:new.t_id where T_ID =:old.t_id;
update subjects set T_ID=:new.t_id where T_ID =:old.t_id;
end;

update TEACHER set t_id='PMT013' where t_id='PMT001';
```

Script Output × | Query Result ×

Task completed in 0.294 seconds

```
Trigger TRG_AF_UP_T compiled

1 row updated.
```

Worksheet | Query Builder

```
/*Triggers*/
create or replace trigger trg_af_up_t
after update of t_id on TEACHER
referencing new as new old as old
for each row
begin
update Class set t_id=:new.t_id where t_id =:old.t_id;
update Exam set T_ID=:new.t_id where T_ID =:old.t_id;
update Homework set T_ID=:new.t_id where T_ID =:old.t_id;
update marks set T_ID=:new.t_id where T_ID =:old.t_id;
update subjects set T_ID=:new.t_id where T_ID =:old.t_id;
end;

update TEACHER set t_id='PMT013' where t_id='PMT001';

select * from exam;
```

Script Output × | Query Result ×

SQL | All Rows Fetched: 4 in 0.088 seconds

	EXAM_CODE	EXAM_SUB	EXAM_DATE	EXAM_QPAPER	T_ID
1	EXM001	PHY	08-06-20	qpaper1.docx	PMT013
2	EXM003	CHEM	05-06-20	qpaper3.docx	PMT005
3	EXM004	BENG	04-06-20	qpaperx.docx	TMP004
4	EXM005	CHML	03-06-20	qpaper5.docx	PMT005

Worksheet | Query Builder

```
/*Triggers*/
create or replace trigger trg_af_up_t
after update of t_id on TEACHER
referencing new as new old as old
for each row
begin
update Class set t_id=:new.t_id where t_id =:old.t_id;
update Exam set T_ID=:new.t_id where T_ID =:old.t_id;
update Homework set T_ID=:new.t_id where T_ID =:old.t_id;
update marks set T_ID=:new.t_id where T_ID =:old.t_id;
update subjects set T_ID=:new.t_id where T_ID =:old.t_id;
end;

update TEACHER set t_id='PMT013' where t_id='PMT001';

select * from exam;
select * from class;
```

Script Output × | Query Result ×

SQL | All Rows Fetched: 5 in 0.002 seconds

	STD_SEC	AD_ID	T_ID	TOTAL_STUDENTS
1	12A	3	PMT013	33
2	12C	4	TMP002	28
3	12F	4	PMT003	32
4	11A	5	TMP004	30
5	11F	3	PMT005	28

Worksheet | Query Builder

```
/*Triggers*/
create or replace trigger trg_af_up_t
after update of t_id on TEACHER
referencing new as new old as old
for each row
begin
update Class set t_id=:new.t_id where t_id =:old.t_id;
update Exam set T_ID=:new.t_id where T_ID =:old.t_id;
update Homework set T_ID=:new.t_id where T_ID =:old.t_id;
update marks set T_ID=:new.t_id where T_ID =:old.t_id;
update subjects set T_ID=:new.t_id where T_ID =:old.t_id;
end;

update TEACHER set t_id='PMT013' where t_id='PMT001';

select * from exam;
select * from class;
select * from marks;
```

Script Output × | Query Result ×

SQL | All Rows Fetched: 8 in 0.041 seconds

	PAPER_NO	MARKS_OBT	TOTAL_MARKS	SUBJECT	STUDENT_NAME	T_ID	ROLL_NO
1	SL001	98	100	PHYSICS	Sahasra	PMT003	14
2	SL0001	97	100	PHYSICS	Sree	PMT003	26
3	SL003	91	100	PHYSICS	Uday	PMT003	12
4	SL004	92	100	CHEMISTRY	Sahasra	PMT013	14
5	SL005	90	100	CHEMISTRY	Sree	PMT013	26
6	SL018	0	100	MATHS	Akhil	TMP002	21
7	SL024	0	100	HISTORY	Abhijeet	TMP004	5
8	SL006	0	100	GEOGRAPHY	Akhil	TMP004	21

Worksheet | Query Builder

```
/*Triggers*/
create or replace trigger trg_af_up_t
after update of t_id on TEACHER
referencing new as new old as old
for each row
begin
update Class set t_id=:new.t_id where t_id =:old.t_id;
update Exam set T_ID=:new.t_id where T_ID =:old.t_id;
update Homework set T_ID=:new.t_id where T_ID =:old.t_id;
update marks set T_ID=:new.t_id where T_ID =:old.t_id;
update subjects set T_ID=:new.t_id where T_ID =:old.t_id;
end;

update TEACHER set t_id='PMT013' where t_id='PMT001';

select * from exam;
select * from class;
select * from marks;
select * from subjects;
```

Script Output × | Query Result ×

SQL | All Rows Fetched: 5 in 0.085 seconds

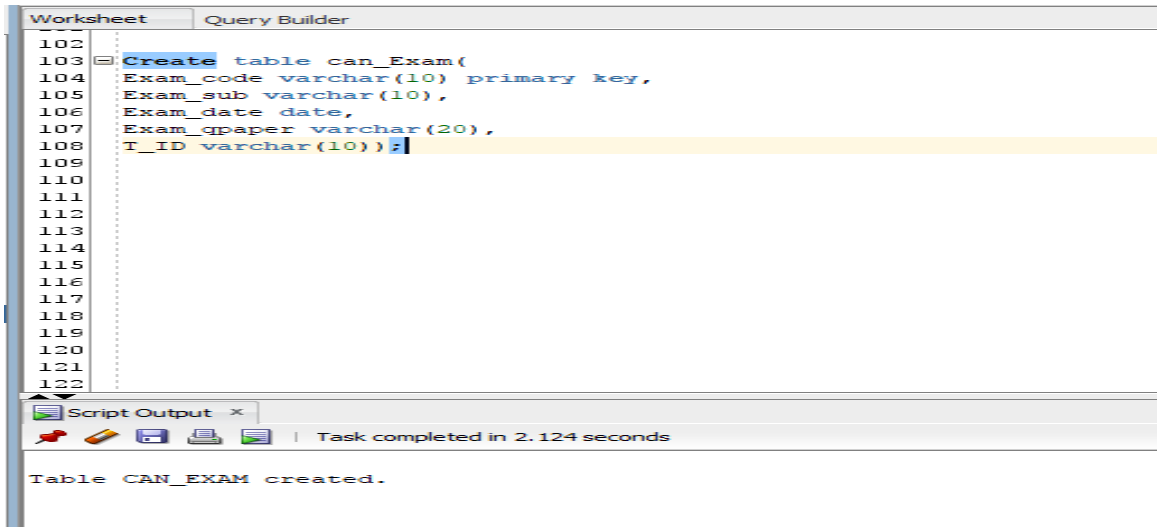
	SUBJ_CODE	NAME	T_ID	TOT_STUD_ENROL
1	PHY1001	Introduction to Physics	PMT003	70
2	CHY1001	Introduction to Chemistry	PMT013	35
3	BENG1001	Bengali for Beginners	TMP004	80
4	HIST1001	History In Brief	TMP002	75
5	PHY1018	Instrumental Physics	PMT003	65

6.WHEN A EXAM IS CANCLED, RECORD OF CANCELLED EXAMS IS NOTED IN A TABLE

```
Create table can_Exam(
Exam_code varchar(10) primary key,
Exam_sub varchar(10),
Exam_date date,
Exam_qpaper varchar(20),
T_ID varchar(10));

create or replace trigger exm_det
after delete on Exam
referencing new as new old as old
for each row
begin
insert into can_Exam
values(:old.Exam_code,:old.Exam_sub,:old.Exam_date,:old.Exam_qpaper,:old.T_ID);
dbms_output.put_line('the exam is cancelled' || :old.Exam_code);
end;

delete Exam where Exam_code='EXM004';
delete exam where exam_code='EXM003';
select * from can_exam;
```



Worksheet | Query Builder

```
Create table can_Exam(
Exam_code varchar(10) primary key,
Exam_sub varchar(10),
Exam_date date,
Exam_qpaper varchar(20),
T_ID varchar(10));

create or replace trigger exm_det
after delete on Exam
referencing new as new old as old
for each row
begin
insert into can_Exam
values(:old.Exam_code,:old.Exam_sub,:old.Exam_date,:old.Exam_qpaper,:old.T_ID);
dbms_output.put_line('the exam is cancelled' || :old.Exam_code);
end;
```

Script Output | Task completed in 0.323 seconds

```
Table CAN_EXAM created.


Trigger EXM_DET compiled
```

Worksheet | Query Builder

```
Create table can_Exam(
Exam_code varchar(10) primary key,
Exam_sub varchar(10),
Exam_date date,
Exam_qpaper varchar(20),
T_ID varchar(10));

create or replace trigger exm_det
after delete on Exam
referencing new as new old as old
for each row
begin
insert into can_Exam
values(:old.Exam_code,:old.Exam_sub,:old.Exam_date,:old.Exam_qpaper,:old.T_ID);
dbms_output.put_line('the exam is cancelled' || :old.Exam_code);
end;

delete Exam where Exam_code='EXM004';
delete exam where exam_code='EXM003';

select * from can_exam;
```

Script Output | Query Result

SQL | All Rows Fetched: 2 in 0.001 seconds

	EXAM_CODE	EXAM_SUB	EXAM_DATE	EXAM_QPAPER	T_ID
1	EXM004	BENG	04-06-20	qpaperx.docx	TMP004
2	EXM003	CHEM	05-06-20	qpaper3.docx	PMT005

7.WHEN A STUDENT ROLL NUMBER IS CHANGED RELATED DATA IS TO BE UPDATED

```
create or replace trigger trg_stud_upd
after update of roll_no on student
referencing new as new old as old
for each row
begin
update marks set Roll_No=:new.roll_No where Roll_No=:old.roll_no;
update parent set roll_no=:new.roll_no where roll_no=:old.roll_no;
end;

update student set roll_no=50 where roll_no=14;
select * from parent;
select * from marks;
```

Worksheet | Query Builder

```
create or replace trigger trg_stud_upd
after update of roll_no on student
referencing new as new old as old
for each row
begin
update marks set Roll_No=:new.roll_No where Roll_No=:old.roll_no;
update parent set roll_no=:new.roll_no where roll_no=:old.roll_no;
end;

update student set roll_no=50 where roll_no=14;
select * from parent;
select * from marks;
```

Script Output | Task completed in 0.234 seconds

```
Trigger TRG_STUD_UPD compiled


1 row updated.
```

Worksheet | Query Builder

```
create or replace trigger trg_stud_upd
after update of roll_no on student
referencing new as new old as old
for each row
begin
update marks set Roll_No=:new.roll_No where Roll_No=:old.roll_no;
update parent set roll_no=:new.roll_no where roll_no=:old.roll_no;
end;

update student set roll_no=50 where roll_no=14;
select * from parent;
select * from marks;
```

Script Output | Query Result

SQL | All Rows Fetched: 5 in 0.002 seconds

	ROLL_NO	PARENT_NAME	PHONE_NO
1	50	BadriNath	9912342395
2	12	Lakshmi	9898090990
3	21	Raghuram	8989898989
4	5	Koteswar Rao	9080908090
5	26	Kusuma	9080990099

Worksheet | Query Builder

```
create or replace trigger trg_stud_upd
after update of roll_no on student
referencing new as new old as old
for each row
begin
update marks set Roll_No=:new.roll_No where Roll_No=:old.roll_no;
update parent set roll_no=:new.roll_no where roll_no=:old.roll_no;
end;

update student set roll_no=50 where roll_no=14;
select * from parent;
select * from marks;
```

Script Output | Query Result

SQL | All Rows Fetched: 8 in 0.022 seconds

	PAPER_NO	MARKS_OBT	TOTAL_MARKS	SUBJECT	STUDENT_NAME	T_ID	ROLL_NO
1	SL001	98	100	PHYSICS	Sahasra	PMT003	50
2	SL0001	97	100	PHYSICS	Sree	PMT003	26
3	SL003	91	100	PHYSICS	Uday	PMT003	12
4	SL004	92	100	CHEMISTRY	Sahasra	PMT013	50
5	SL005	90	100	CHEMISTRY	Sree	PMT013	26
6	SL018	0	100	MATHS	Akhil	TMP002	21
7	SL024	0	100	HISTORY	Abhijeet	TMP004	5
8	SL006	0	100	GEOGRAPHY	Akhil	TMP004	21

__THANK YOU__