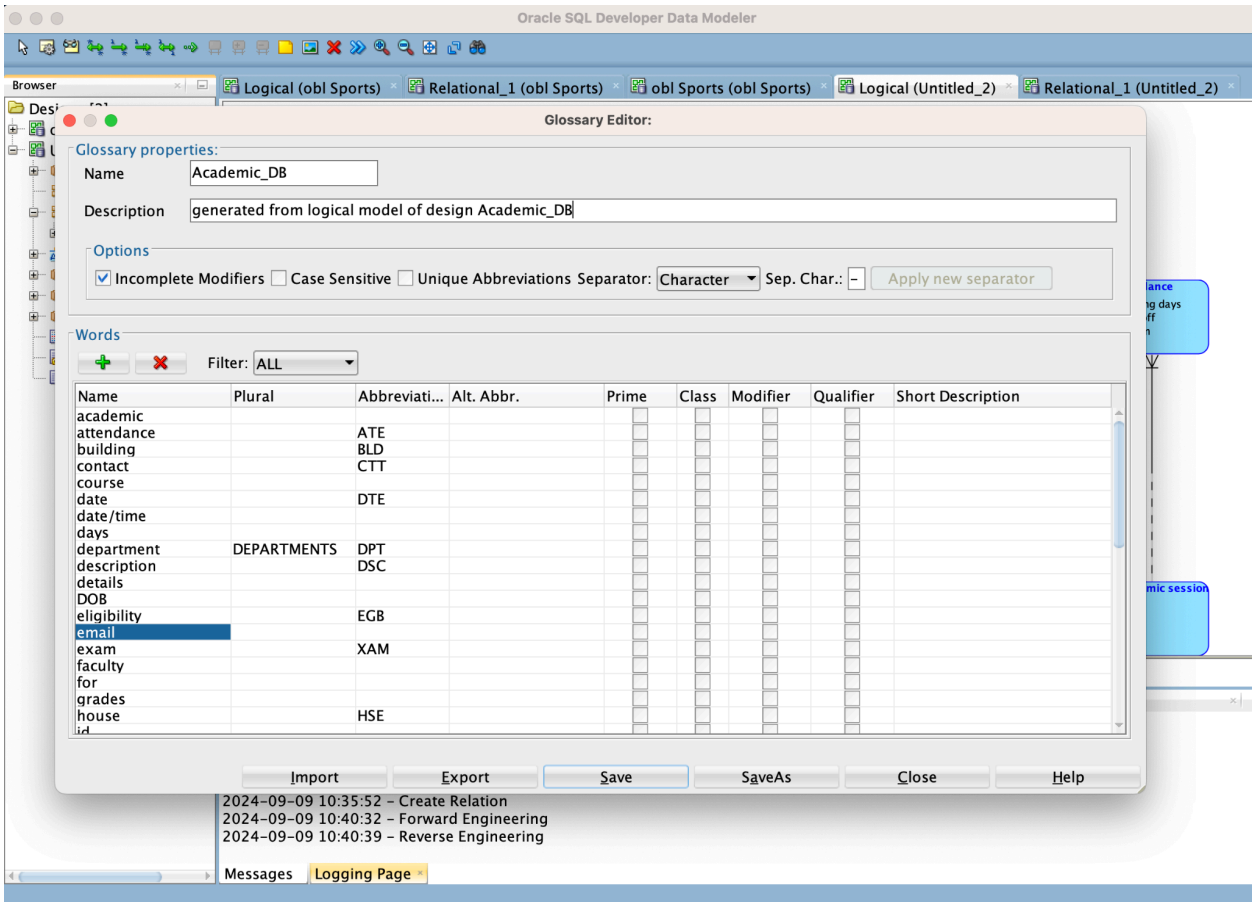
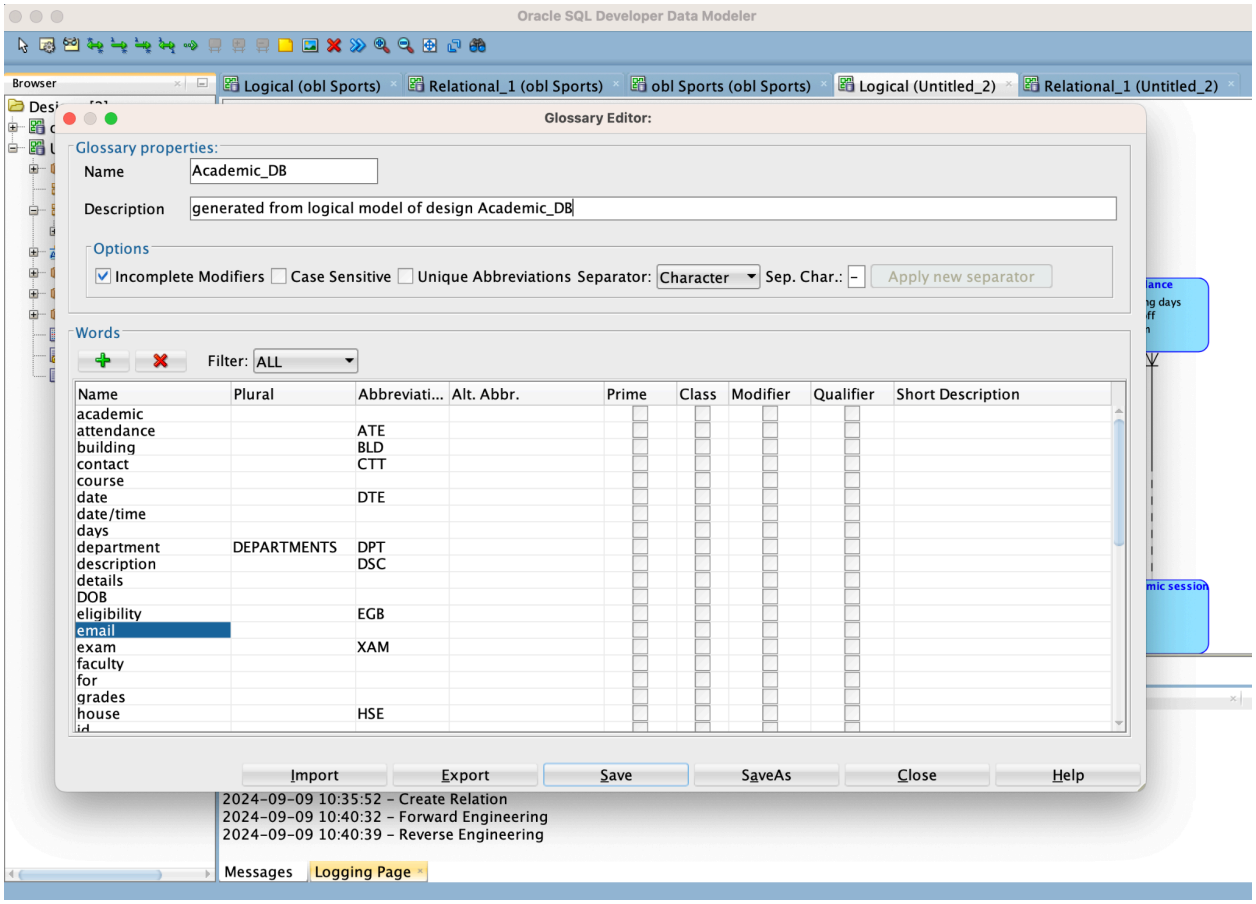
Mint

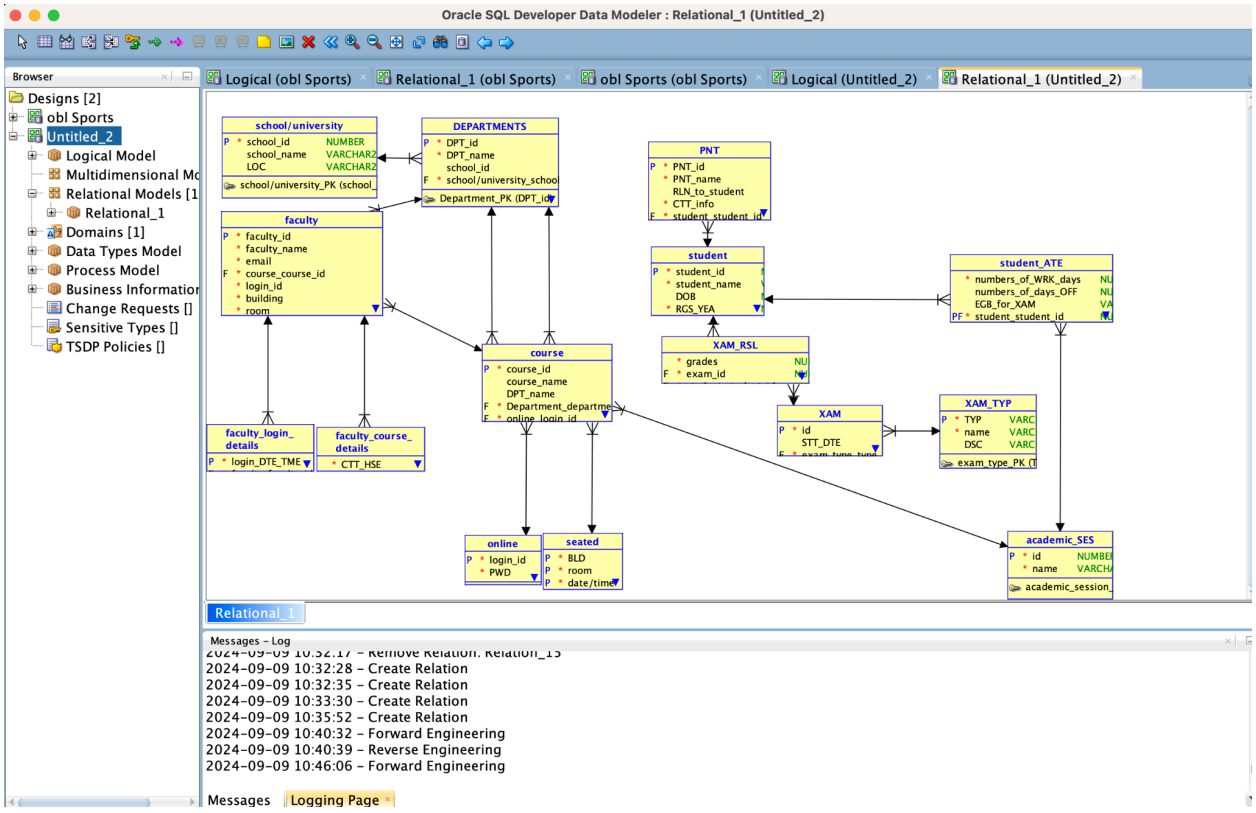
Foundation 5 Practice

Exercise 1: Creating a glossary from a logical model



Exercise 2: Forward engineering design to apply Glossary and Naming Standard


Oracle SQL Developer Data Modeler
Logical (obl Sports)
Relational_1 (obl Sports)
obl Sports (obl Sports)
Logical (Untitled_2)
Relational_1 (Untitled_2)
Glossary Editor:
Glossary properties:
Name
Academic_DB
Description
generated from logical model of design Academic_DB
Options
Incomplete Modifiers
Case Sensitive
Unique Abbreviations
Separator:
Character
Sep. Char.:
Apply new separator
Words
Filter:
ALL
Name
Plural
Abbreviati...
Alt. Abbr.
Prime
Class
Modifier
Qualifier
Short Description
academic
attendance
ATE
building
BLD
contact
CTT
course
date
DTE
date/time
days
department
DEPARTMENTS
DPT
description
DSC
details
DOB
eligibility
EGB
email
exam
XAM
faculty
for
grades
house
HSE
Import
Export
Save
SaveAs
Close
Help
2024-09-09 10:35:52 – Create Relation
2024-09-09 10:40:32 – Forward Engineering
2024-09-09 10:40:39 – Reverse Engineering
Messages
Logging Page


Oracle SQL Developer Data Modeler : Relational_1 (Untitled_2)
Browser
Designs [2]
obl Sports
Untitled_2
Logical Model
Multidimensional Mo
Relational Models [1
Relational_1
Domains [1]
Data Types Model
Process Model
Business Informatior
Change Requests []
Sensitive Types []
TSDP Policies []
Logical (obl Sports)
Relational_1 (obl Sports)
obl Sports (obl Sports)
Logical (Untitled_2)
Relational_1 (Untitled_2)
school/university
DEPARTMENTS
PNT
faculty
student
student_ATE
course
XAM_RSL
XAM
XAM_TYP
faculty_login_details
faculty_course_details
online
seated
academic_SES
Relational_1
Messages – Log
2024-09-09 10:32:28 – Create Relation
2024-09-09 10:32:35 – Create Relation
2024-09-09 10:33:30 – Create Relation
2024-09-09 10:35:52 – Create Relation
2024-09-09 10:40:32 – Forward Engineering
2024-09-09 10:40:39 – Reverse Engineering
2024-09-09 10:46:06 – Forward Engineering
Messages
Logging Page

Foundation 6 Practice

6-1: Getting familiar with Oracle Apex
6-2: Getting familiar with Application Express SQl Workshop: Managing db objects with Object Browser, Using SQL Commands, Using SQL Scripts
6-3:

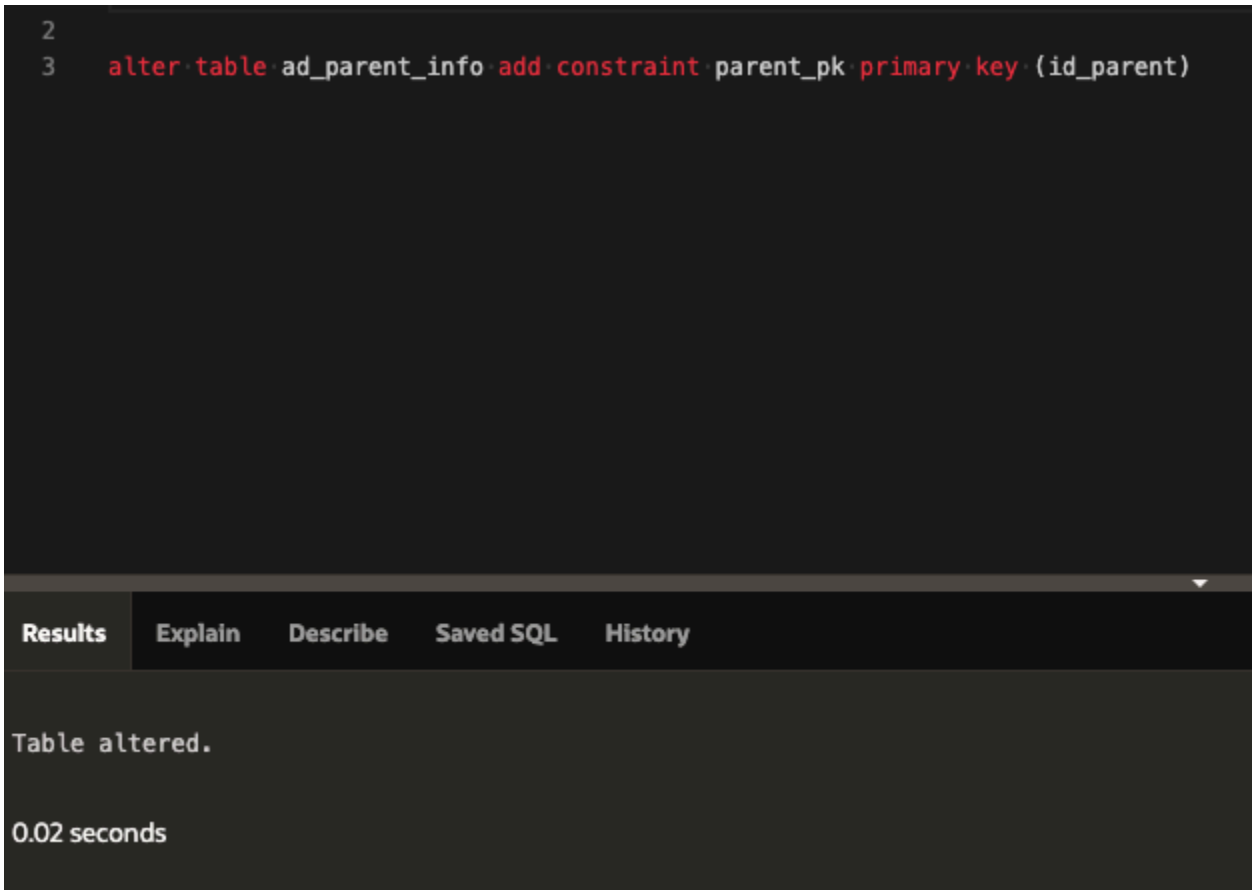
Exercise 1: create table using oracle APEX. My code was the following:

```
CREATE TABLE ad_STUDENT(
   id_student number(10) not null,
   name_first varchar2(20) not null,
   last_name varchar2(20) not null,
   reg_year date not null
);

create table ad_parent_info(
   id_parent varchar2(50) not null,
   name_first_parent1 varchar(20) not null,
   name_last_parent1 varchar(20) not null,
   name_first_parent2 varchar(20) not null,
   name_last_parent2 varchar(20) not null
);

create table ad_student_course(
   id_course varchar2(10) not null,
   name_course varchar2(20) not null
);

create table ad_course_online(
   id_login_course_online varchar2(50) not null,
   password_course_online varchar2(50) not null
);

create table ad_course_seated(
   building_course_seated varchar2(20) not null,
   room_course_seated varchar2(10) not null,
   date_time_course_seated date not null
)

create table ad_student attendance(
   number_workingdays number(3) not null,
   number_daysoff number(3) not null,
   eligbility_exam char(3) not null
```

```
);

create table ad_academic_session(
    id_academicsession char(10) not null,
    name_academicsession char(10) not null
);

create table ad_department(
    id_department number(10) not null,
    name_department char(20) not null,
    head_department char(100) not null
);

create table ad_faculty(
    id_faculty varchar2(50) not null,
    name_first varchar2(20) not null,
    name_last varchar2(20) not null,
    email varchar2(50) not null
)

create table ad_faculty_course_detail(
    hours_contact number (5) not null
);

create table ad_faculty_login_detail(
    DT_login date not null
);

create table ad_exam_result(
    grade number(3) not null
);

create table ad_exam(
    id_exam number(10) not null,
    date_start date not null
);

create table ad_exam_type(
    type_exam varchar2(10) not null,
    name_exam varchar2(10) not null,
    description_exam varchar2 (100) not null
);
```

----Table was created succcesfully--

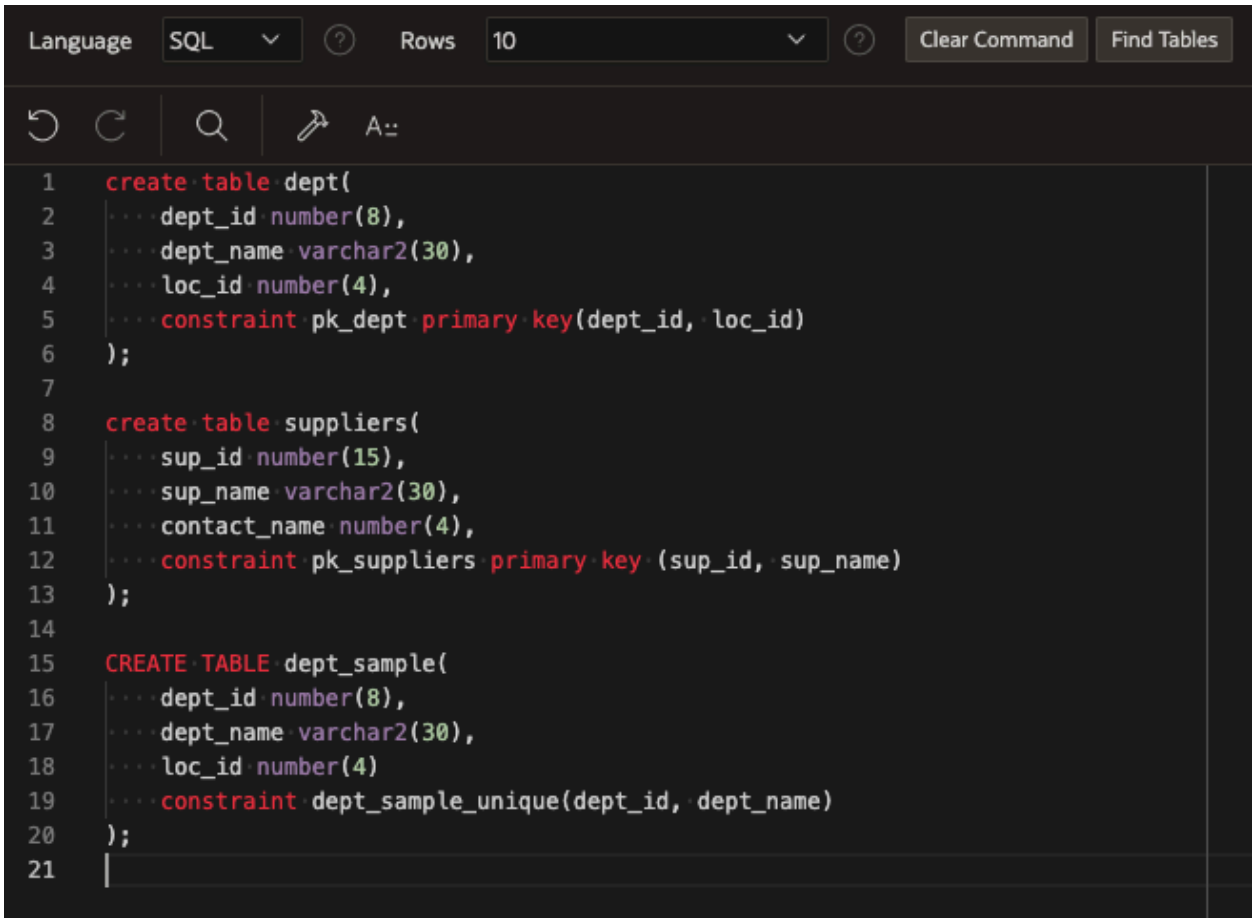
Exercise 2:Altering the Tables: The following codes were used
alter table ad_STUDENT add constraint student_pk primary key (id_student);
alter table ad_parent_info add constraint parent_pk primary key (id_parent);
alter table ad_academic_session add constraint academicsession_pk primary key (id_academicsession);
alter table ad_student_course add constraint course_pk primary key (id_course);
alter table ad_department add constraint department_pk primary key (id_department);
alter table ad_exam add constraint exam_pk primary key (id_exam);
alter table ad_exam_type add constraint examtype_pk primary key (type_exam);
alter table ad_faculty add constraint faculty_pk primary key (id_faculty);

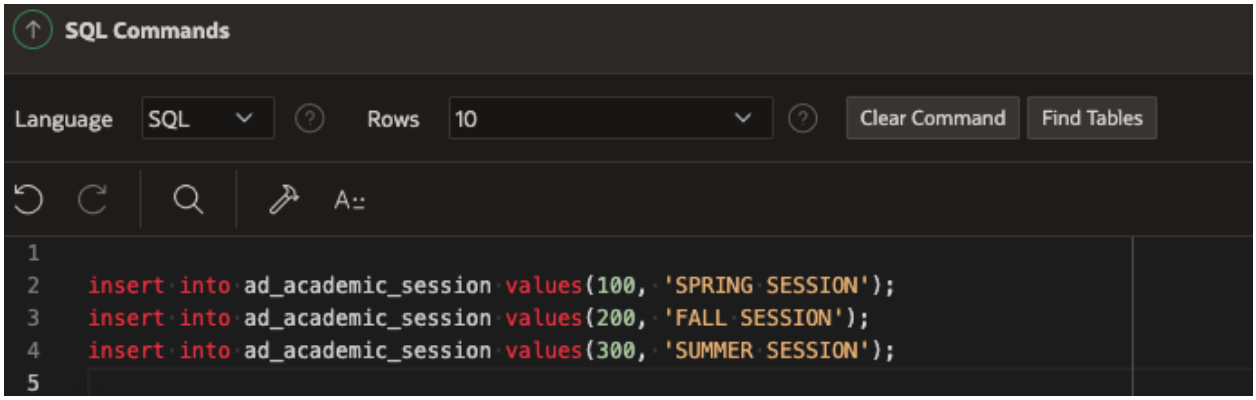


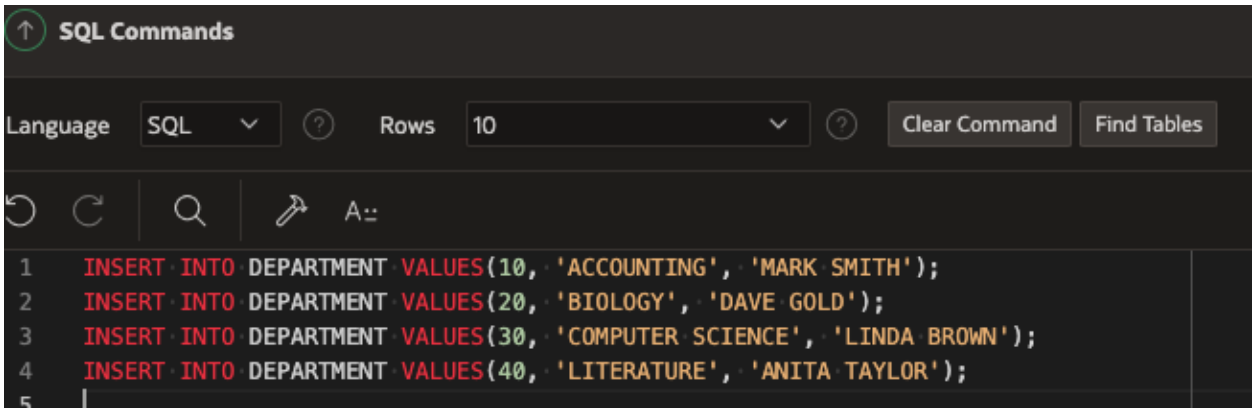
Exercise 3:

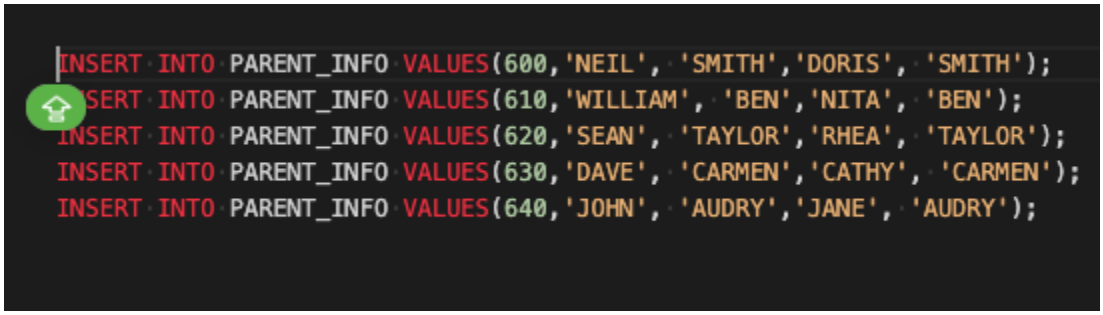
```
create table dept(
   dept_id number(8),
   dept_name varchar2(30),
   loc_id number(4),
```

```
    constraint pk_dept primary key(dept_id, loc_id)
)

create table suppliers(
    sup_id number(15),
    sup_name varchar2(30),
    contact_name number(4),
    constraint pk_suppliers primary key (sup_id, sup_name)
)

CREATE TABLE dept_sample(
    dept_id number(8),
    dept_name varchar2(30),
    loc_id number(4)
    constraint dept_sample_unique(dept_id, dept_name)
)
```









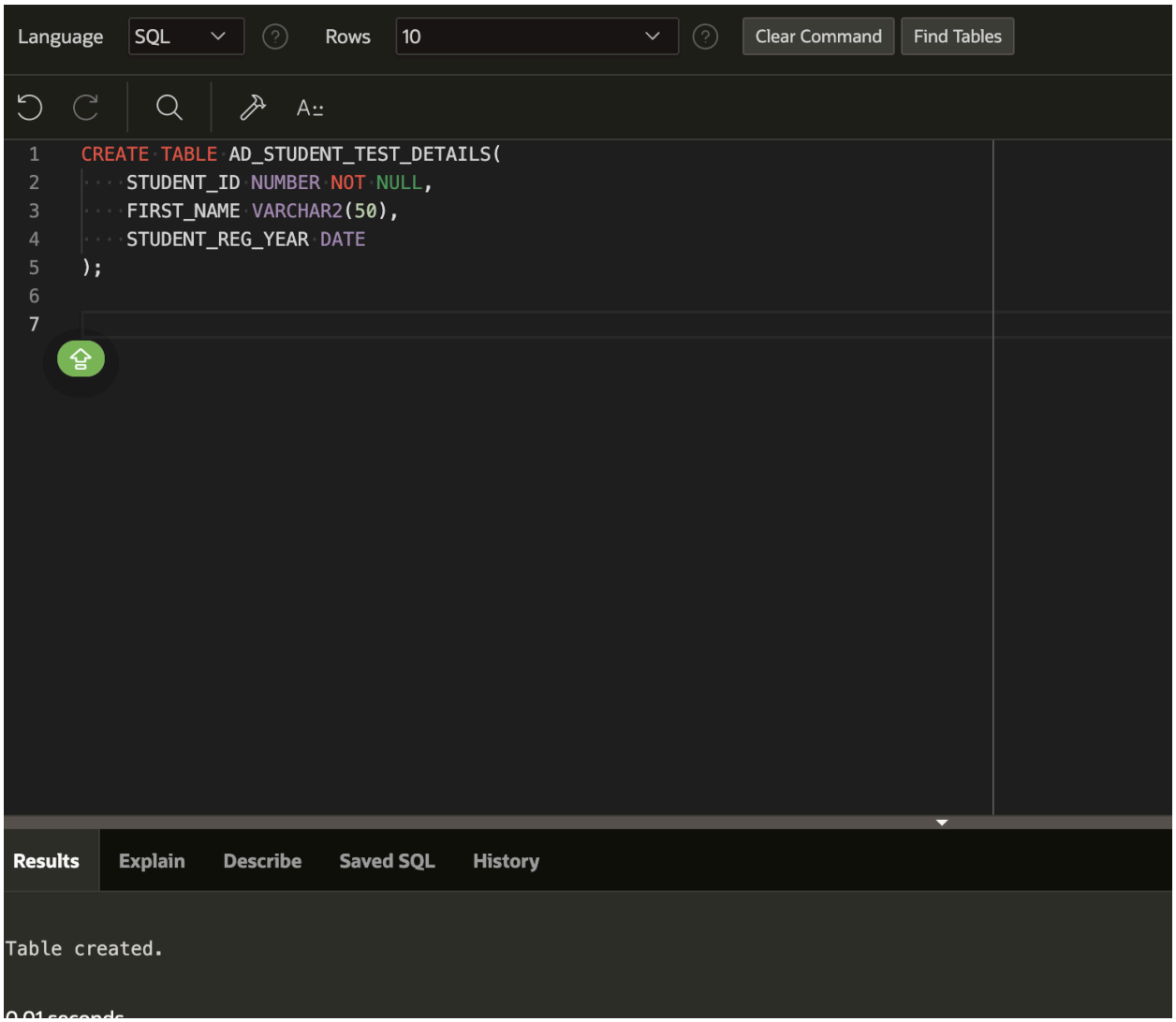
INSERT INTO STUDENT VALUES(720, 'JACK','SMITH',
TO_DATE('01-Jan-2012','DD-Mon-YYYY'),'JSMITH@SCHOOL.EDU', 600);
INSERT INTO STUDENT VALUES(730, 'NOAH', 'AUDRY',
TO_DATE('01-Jan-2012','DD-Mon-YYYY'),'NAUDRY@SCHOOL.EDU', 640);
INSERT INTO STUDENT VALUES(740, 'RHONDA','TAYLOR',
TO_DATE('01-Sep-2012','DD-Mon-YYYY'),'RTAYLOR@SCHOOL.EDU', 620);
INSERT INTO STUDENT VALUES(750, 'ROBERT','BEN',
TO_DATE('01-Mar-2012','DD-Mon-YYYY'),'RBEN@SCHOOL.EDU', 610);
INSERT INTO STUDENT VALUES(760, 'JEANNE','BEN',
TO_DATE('01-Mar-2012','DD-Mon-YYYY'),'JBEN@SCHOOL.EDU', 610);
INSERT INTO STUDENT VALUES(770, 'MILLS','CARMEN',
TO_DATE('01-Apr-2013','DD-Mon-YYYY'),'MCARMEN@SCHOOL.EDU', 630);

```
INSERT INTO COURSE VALUES (190, 'PRINCIPLES OF ACCOUNTING', 100,
10,NULL,NULL,'BUILDING A','101','MWF 12-1');
INSERT INTO COURSE VALUES (191, 'INTRODUCTION TO BUSINESS LAW', 100,
10,NULL,NULL,'BUILDING B','201','THUR 2-4');
INSERT INTO COURSE VALUES (192, 'COST ACCOUNTING', 100,
10,NULL,NULL,'BUILDING C','301','TUES 5-7');
INSERT INTO COURSE VALUES (193, 'STRATEGIC TAX PLANNING FOR BUSINESS', 100,
10,'TAX123','PASSWORD',NULL,NULL,NULL);
INSERT INTO COURSE VALUES (194, 'GENERAL BIOLOGY', 200,
20,'BIO123','PASSWORD',NULL,NULL,NULL);
INSERT INTO COURSE VALUES (195, 'CELL BIOLOGY', 200, 20,NULL,NULL,'BUILDING
D','401','MWF 9-10');

INSERT INTO FACULTY VALUES (800, 'JILL',
'MILLER','JMILL@SCHOOL.EDU',10000,'HEALTH',NULL,20);
INSERT INTO FACULTY VALUES (810, 'JAMES',
'BORG','JBORG@SCHOOL.EDU',30000,'HEALTH,DENTAL',NULL,10);
INSERT INTO FACULTY VALUES (820, 'LYNN',
'BROWN','LBROWN@SCHOOL.EDU',NULL,NULL,50,30);
INSERT INTO FACULTY VALUES (830, 'ARTHUR',
'SMITH','ASMITH@SCHOOL.EDU',NULL,NULL,40,10);
INSERT INTO FACULTY VALUES (840, 'SALLY',
'JONES','SJONES@SCHOOL.EDU',50000,'HEALTH,DENTAL,VISION',NULL,40);

INSERT INTO EXAM_TYPE VALUES('MCE','Multiple Choice Exams','CHOOSE MORE THAN
ONE ANSWER');
INSERT INTO EXAM_TYPE VALUES('TF','TRUE AND FALSE Exams','CHOOSE EITHER
TRUE OR FALSE');
INSERT INTO EXAM_TYPE VALUES('FIB','FILL IN THE BLANKS Exams','TYPE IN THE
CORRECT ANSWER');
INSERT INTO EXAM_TYPE VALUES('ESS','ESSAY Exams','WRITE PARAGRAPHS');
INSERT INTO EXAM_TYPE VALUES('SA','SHORT ANSWER Exams','WRITE SHORT
ANSWERS');

INSERT INTO EXAM VALUES(500, TO_DATE('12-Sep-2013','DD-Mon-YYYY'),'MCE',190);
INSERT INTO EXAM VALUES(510, TO_DATE('15-Sep-2013','DD-Mon-YYYY'),'SA', 191);
INSERT INTO EXAM VALUES(520, TO_DATE('18-Sep-2013','DD-Mon -YYYY'),'FIB', 192);
INSERT INTO EXAM VALUES(530, TO_DATE('21-Mar-2014','DD-Mon -YYYY'),'ESS', 193);
INSERT INTO EXAM VALUES(540, TO_DATE('02-Apr-2014','DD-Mon-YYYY'),'TF', 194);

INSERT INTO EXAM_RESULT VALUES(720,190,500,91);
INSERT INTO EXAM_RESULT VALUES(720,193,520,97);
INSERT INTO EXAM_RESULT VALUES(730,195,540,87);
```

```
INSERT INTO EXAM_RESULT VALUES(730,194,530,85);
INSERT INTO EXAM_RESULT VALUES(750,192,500,60);
INSERT INTO EXAM_RESULT VALUES(750,195,510,97);
INSERT INTO EXAM_RESULT VALUES(750,191,520,78);
INSERT INTO EXAM_RESULT VALUES(760,192,540,65);
INSERT INTO EXAM_RESULT VALUES(760,191,530,60);
INSERT INTO EXAM_RESULT VALUES(760,192,510,70);

INSERT INTO STUDENT_ATTENDANCE VALUES( 720,100, 180, 21,'Y');
INSERT INTO STUDENT_ATTENDANCE VALUES( 730,200, 180, 11,'Y');
INSERT INTO STUDENT_ATTENDANCE VALUES( 740,300, 180, 12,'Y');
INSERT INTO STUDENT_ATTENDANCE VALUES( 750,100, 180, 14,'Y');
INSERT INTO STUDENT_ATTENDANCE VALUES( 760,200, 180, 15,'Y');
INSERT INTO STUDENT_ATTENDANCE VALUES( 770,300, 180, 13,'Y');

INSERT INTO STUDENT_COURSE_DET VALUES(720,190,'A');
INSERT INTO STUDENT_COURSE_DET VALUES(720,193,'B');
INSERT INTO STUDENT_COURSE_DET VALUES(730,191,'C');
INSERT INTO STUDENT_COURSE_DET VALUES(740,195,'F');
INSERT INTO STUDENT_COURSE_DET VALUES(750,192,'A');
INSERT INTO STUDENT_COURSE_DET VALUES(760,190,'B');
INSERT INTO STUDENT_COURSE_DET VALUES(760,192,'C');
INSERT INTO STUDENT_COURSE_DET VALUES(770,192,'D');
INSERT INTO STUDENT_COURSE_DET VALUES(770,193,'F');
INSERT INTO STUDENT_COURSE_DET VALUES(770,194,'A');

INSERT INTO FACULTY_COURSE_DETAIL VALUES (800, 192,3);
INSERT INTO FACULTY_COURSE_DETAIL VALUES (800, 193,4);
INSERT INTO FACULTY_COURSE_DETAIL VALUES (800, 190,5);
INSERT INTO FACULTY_COURSE_DETAIL VALUES (800, 191,3);
INSERT INTO FACULTY_COURSE_DETAIL VALUES (810, 194,4);
INSERT INTO FACULTY_COURSE_DETAIL VALUES (810, 195,5);

INSERT INTO FACULTY_LOGIN_DETAIL VALUES(800, CURRENT_TIMESTAMP);
INSERT INTO FACULTY_LOGIN_DETAIL VALUES(810, CURRENT_TIMESTAMP);
INSERT INTO FACULTY_LOGIN_DETAIL VALUES(840, CURRENT_TIMESTAMP);
INSERT INTO FACULTY_LOGIN_DETAIL VALUES(820, CURRENT_TIMESTAMP);
INSERT INTO FACULTY_LOGIN_DETAIL VALUES(830, CURRENT_TIMESTAMP);


ALTER TABLE FACULTY_LOGIN_DETAIL ADD  DETAILS  VARCHAR2(50);
UPDATE FACULTY_LOGIN_DETAIL
SET DETAILS = 'hello'
Where FACULTY_LOGIN_DETAIL.ID = 1
```

UPDATE FACULTY_LOGIN_DETAIL
SET DETAILS = 'hola
Where FACULTY_LOGIN_DETAIL.ID = 2

6-5



CREATE TABLE AD_STUDENT_TEST_DETAILS
(
STUDENT_ID NUMBER NOT NULL,
FIRST_NAME VARCHAR2(50),
STUDENT_REG_YEAR DATE
) ;

ALTER TABLE AD_STUDENT_TEST_DETAILS ADD (EMAIL_ADDR VARCHAR2(100)
UNIQUE);

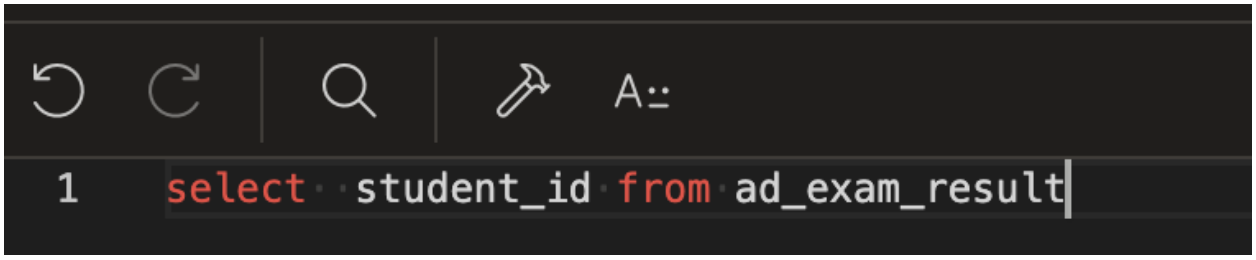
SAVEPOINT ALTER_DONE;

ROLLBACK TO ALTER_DONE;

- The new email field will still be there after the rollback because changes are made after the save point
- After DELETE: insert would have added the rows in the test table and also updated. Some of the rows would be deleted. When the rollback is done to Savepoint UPDATE_Done. The delete operation would be undone. The table would be affected by the insert and update, but not by the delete.

6-6

Select * from *



select student_id from ad_exam_result
select exam_eligibility from ad_exam_result
select student_id from ad_student_attendance where exam_eligibility
Select login_date_time from ad_faculty_login_details
Select id_students from ad_student_attendance
Select head from ad_departments
Select id_student ||' :FIRST NAME IS ' || FIRST_NAME AS STUDENT_INFO from ad_students
Select distinct type from ad_exams

6-7
Exercise 1

SELECT course_nameFROM ad_courses WHERE id_session= 100;

SELECT id_student FROM ad_exam_results WHERE GRADE > 95;

SELECT id_student FROM ad_exam_results WHERE GRADE BETWEEN 65 AND 70;

```
SELECT name_first, name_name, reg_year  FROM ad_students WHERE reg_year >
'01-JUNE-2012';

  select HEAD From AD_DEPARTMENTS
  select * From ad_courses WHERE DEPT_ID = 10 or DEPT_ID - 30
  select * From ad_courses WHERE (DEPT_ID = 30 and SESSION_ID = 200)
  select * From ad_courses WHERE (DEPT_ID = 30 and SESSION_ID = 200)
  select * From ad_courses WHERE DEPT_ID = 20
```

6-8

Exercise 1

```
  select * From AD_STUDENTS ORDER BY REG_YEAR
  select * From AD_EXAM_RESULTS ORDER BY (STUDENT_ID, COURSE_ID)
  select * From AD_STUDENT_ATTENDANCE ORDER BY STUDENT_ID
  select * From AD_DEPARTMENTS ORDER BY DEPT_ID
  select * From AD_EXAM_RESULTS ORDER BY EXAM_RESULTS DESC MAX REC
 select * From STUDENT_ID, GRADE FROM AD_EXAM_RESULTS ORDER BY GRADE
LIMIT 5
SELECT * FROM AD_PARENTS ORDER BY PARENT_ID
```

6-9

```
SELECT
        C.COURSE_NAME,
        D.DEPT_NAME
FROM
        AD_COURSES
JOIN
        AD_DEPARTMENTS D ON C.DEPARTMENT_ID = D.DEPARTMENT_ID;

SELECT
        COURSE_NAME
FROM
        AD_COURSES
WHERE
        SESSION_ID = 200;
```

```
	-------

 SELECT
C.COURSE_NAME,
D.DEPT_NAME,
S.STUDENT_NAME
FROM
	AD_COURSES C
JOIN
	AD_DEPARTMENTS D ON C.DEPARTMENT_ID = D.DEPARTMENT_ID
JOIN
	 AD_ENROLLMENTS E ON C.COURSE_ID = E.COURSE_ID
JOIN
	AD_STUDENTS S ON E.STUDENT_ID = S.STUDENT_ID;
 --------
SELECT
	C.COURSE_NAME,
	D.DEPT_NAME, S.STUDENT_NAME
FROM
	AD_COURSES C
JOIN
	AD_DEPARTMENTS D ON C.DEPARTMENT_ID = D.DEPARTMENT_ID
JOIN
	AD_ENROLLMENTS E ON C.COURSE_ID = E.COURSE_ID
JOIN
	AD_STUDENTS S ON E.STUDENT_ID = S.STUDENT_ID WHERE
D.DEPARTMENT_ID = 20;


-----

SELECT
	S.STUDENT_ID, S.STUDENT_NAME
FROM
	AD_EXAM_RESULTS E
JOIN
	AD_STUDENTS S ON E.DEPARTMENT_ID = S.STUDENT_ID
WHERE
	E.COURSE_ID BETWEEN 190 AND 192;
```

```
SELECT
        E.*, C.COURSE_NAME
FROM
        AD_EXAM_RESULTS E
LEFT JOIN AD_COURSES C ON E.COURSE_ID = C.COURSE_ID
```

7: What output would be generated when the given statement is executed?

```
#   SELECT * FROM AD_EXAMS CROSS JOIN AD_EXAM_TYPES;
```

This would combine every row from ad_exams with every row of ad_exam_types.