Department of Computer Science

Project Report, Fall 2021-22

Course	Introduction to Database [Section-C]	Group	07

Group Members:

Student ID	Name	Contribution
20-43923-2	LOVELY AKTER	Introduction
20-43259-1	ATHOY KANTI RAY	Query writing(single raw,multi raw,complex subquery,view,joining-equi,left outer,right outer,full,self join)
20-43961-2	TAWAZ RAHMAN	Query writing(simple query,single raw function,multiple raw function,joining-non-equi join)
20-44121-2	MARINA AFROJ	ER diagram, Normalization, Table create & data insertion

Title:

Student information management system

Instructions:

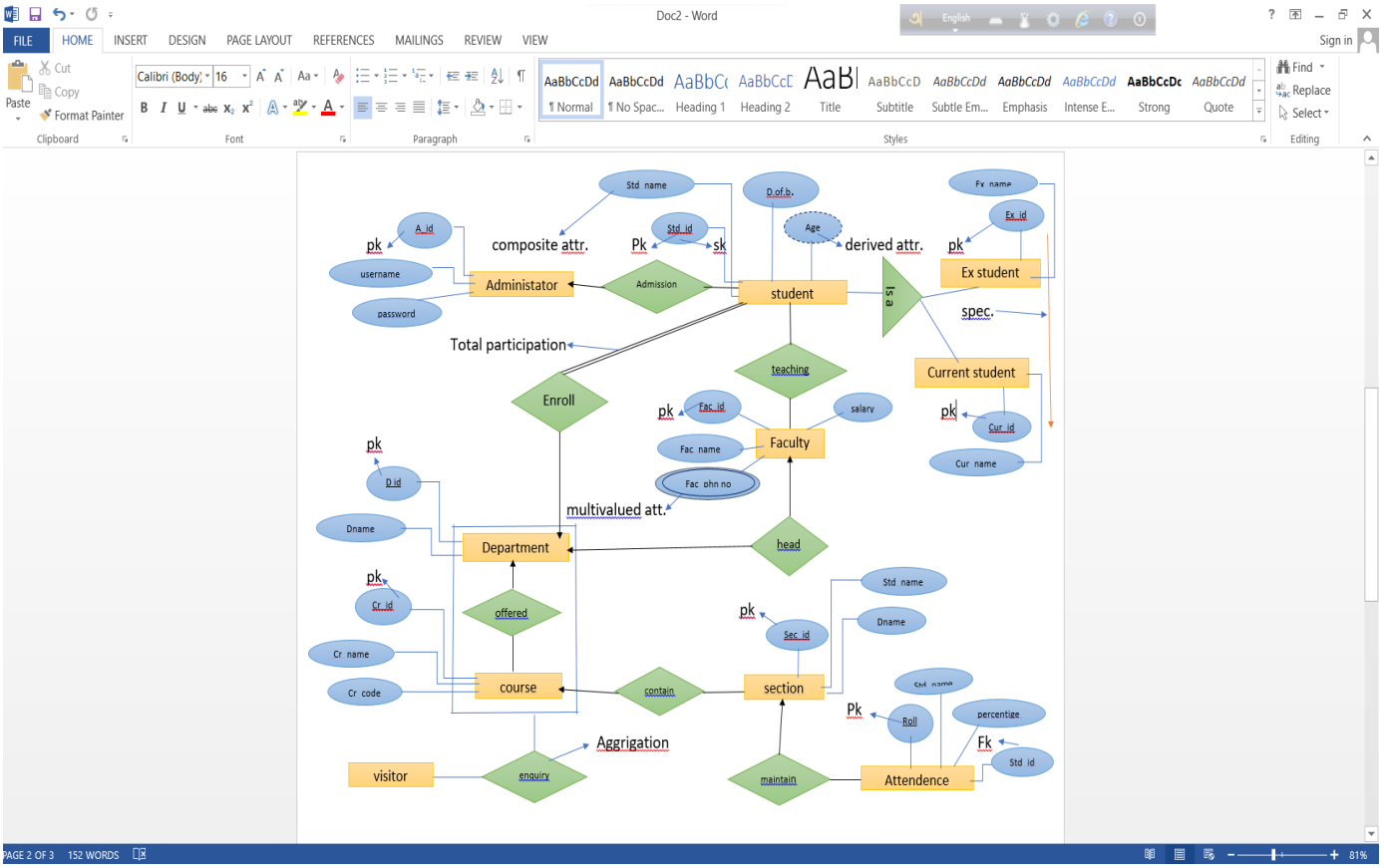
- Fill up *name, id, section, project title.*
- Table Contents
- Introduction (4-5 lines).
- Scenario (descriptive). Give an overview of your project and project objectives. Mention all the **Entities** and **Attributes for each Entity** clearly in your Project Scenario.
- Mid term ERD with Finalized ERD, mention the improvement issues.
- Normalization up to 3NF for each table.
- Perform DDL and DML in Oracle and attach screenshots.
    - Create all the necessary tables with proper constrains after normalization. Take one screenshot after creating one table with create query and describe query -> continue for rest of the tables.

- Insert rows (at least 5) for each table. Take one screenshot after inserting into one table with the insert queries and select all(*) query -> continue for rest of the tables.
- Query (Generate question and provide solution): Do not forget to insert Screenshots.
    - 2 simple queries with simple conditions
    - 1 single row function query
    - 1 multiple row function query
    - 2 single row sub queries
    - 1 multi row sub query
    - 2 complex sub queries
    - 6 joining queries –
        - 1 equijoin query
        - 1 non-equijoin
        - 1 Left outer join
        - 1 Right outer join
        - 1 Full outer join
        - 1 self-join
    - 1 view
    - 1 sequence.

CONTENTS

Introduction: Our project name is “Student Information Management System”. Here, in the ER – Diagram, we can see that, the student has an unique student id, Student Phone Number, Date of Birth, Gender, Student age, Student Name and Address. But inside of Student name and Address both contains First Name, Last Name and Area name, area Code Respectively. Next, Administrator contains admin ID, Password and Username. Administrator helps students to enroll in an institute through Admission. Afterwards, Faculty has his own Faculty ID, Faculty mail id, Faculty name and salary. Basically, faculty teaches all the students. Department contains department name, department head and department ID. Department helps the students to choose the courses. Moreover, Course has unique course name, course id, and course code. At the Department, there are many students and Courses have many sections. Though, Course contain only section. Section has unique section id, section name, student name and department name. Section maintains the regular attendance. Thus, Attendance contains Student ID, roll, student name, and percentage of attendance. This is the overall data of our ER – Diagram.

ER Diagram modify: In ER diagram,we deleted some attributes ,changed design and also added participation



NORMALIZATION

Green → Primary Key

Red → Foreign Key

Admission (std_id, std_name, date of birth, age, A_id, password, username)

1NF– There is no multivalued attribute**.**

1. std_id, std_name, date of birth, age, A_id, password, username

2NF –1. std_id ,std_name, age, date of birth,

2. A_id, password, username

3NF **–** 1. std_id, std_name

2.id, age, date of birth

3.A_id, username, password

Final list from Admission –

1. std_id, std_name
2. id, age, date of birth
3. A_id, username, password

Teaching (std_id, std_name, age, date of birth, Fac_id, Fac_name, Fac_phn no, salary)

1NF – There is one (Fac_phn no) multivalued attribute**.**

1. std_id, std_name, age, date of birth, Fac_id, Fac_name, Fac_phn no, salary

2NF – 1. std_id, std_name, age, date of birth

2. Fac_id, Fac_name, Fac_phn no, salary

3NF –1. std_id, std_name

2.id, age, date of birth

3.Fac_id, Fac_name, phone_no, salary

Final list from teaching –

1.std_id, std_name

2.id, age, date of birth

3.Fac_id, Fac_name, phone_no, salary

Enroll (std_id, std_name, date of birth, age, D_id, D_name)

1NF- There is no multivalued attribute**.**

1. std_id, std_name, date of birth, age, D_id, D_name

2NF- 1. std_id, std_name, age, date of birth

2. D_id, D_name

3NF-

1. std_id, std_name
2. id, age, date of birth
3. D_id, D_name

Final list from Enroll-

1. std_id, std_name, id,D_id
2. id, age, date of birth
3. D_id, D_name

Head (Fac_id, Fac_name, Fac_phn no, salary, D_id, D_name)

1NF- There is one (Fac_phn no) multivalued attribute

1. Fac_id, Fac_name, Fac_phn no, salary, D_id, D_name

2NF- 1.Fac_id, Fac_name, Fac_phn no,salary

2.D_id, D_name

3NF- There is no transitive dependency

1.Fac_id, Fac_name, Fac_phn no,salary

2.D_id, D_name

Final list from Head-

1. Fac_id, Fac_name, salary, D_id
2. Fac_id, Fac_phn no (Composit pk)
3. D_id, D_name

Offered (D_id, D_name, cr_id, cr_name, cr_code)

1NF- There is no multivalued attribute.

1. D_id, D_name, cr_id, cr_name, cr_code

2NF- 1.D_id, D_name.

2.cr_id, cr_name, cr_code

3NF- There is no transitive dependency

1.D_id, D_name.
2.cr_id, cr_name, cr_code

Final list from offered-

1.D_id, D_name.
2.cr_id, cr_name, cr_code

Contain-(cr_id, cr_name, cr_code, sec_id, std_name, D_name)

1NF- There is no multivalued attribute

1. cr_id, cr_name, cr_code, sec_id, std_name, D_name

2NF- 1. cr_id,cr_name,cr_code

2.sec_id,std_name,D_name

3NF- There is no transitive dependency

1.cr_id, cr_name, cr_code

2.sec_id,std_name,D_name

Final list from contain-

1.cr_id,cr_name,cr_code

2.sec_id,std_name,D_name

Maintain (sec_id, D_name, std_name, std_id, Roll, percentige)

1NF – There is no multivalued attribute

1. sec_id, D_name, std_name, std_id, Roll, percentige

2NF- 1.Roll, percentige, std_name, std_id

2. sec_id, std_name, D_name

3NF- There is no transitive dependency

1. Roll, percentige, std_name, std_id
2. sec_id, std_name, D_name

Final list from maintain-

1. Roll, percentige, std_name, std_id
2. sec_id, std_name, D_name

Is a (std_id, std_name, Date of birth, age, Ex_id, Ex_name, cur_id, cur_name)

1NF – There is no multivalued attribute

1. std_id, std_name, Date of birth, age, Ex_id, Ex_name, cur_id, cur_name

2NF- 1. std_id, std_name, Date of birth, age

2.id, age,date of birth

3.Ex_id, Ex_name

4.cur_id, cur_name

3NF- There is no transitive dependency

1. std_id, std_name, Date of birth, age
2. id, age,date of birth
3. Ex_id, Ex_name
4. cur_id, cur_name

Final list from Is a-

1. std_id, std_name, Date of birth, age
2. id, age,date of birth
3. Ex_id, Ex_name
4. cur_id, cur_name

Final Table-

- std_id, std_name, id, A_id → student
- id, age,date of birth → Age
- Fac_id, Fac_name, salary, D_id → Faculty
- Fac_id, Fac_phn no (Composit pk) → Fac
- A_id, username, password → Admin
- D_id, D_name → Department
- cr_id, cr_name, cr_code,D_id → course
- sec_id, std_name, D_name,cr_id → Section
- Roll, percentige, std_name, std_id,sec_id → Attendence
- Ex_id,Ex_name → Ex-student
- Cur_id,cur_name → current-student

TABLE CREATE & DATA INSERTION

STUDENT

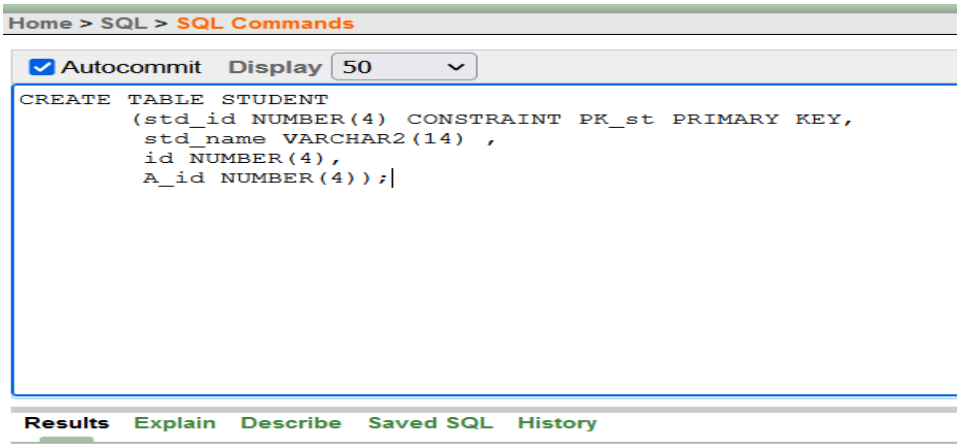
CREATE TABLE STUDENT

(std_id NUMBER(4) CONSTRAINT PK_st PRIMARY KEY,

std_name VARCHAR2(14) ,

id NUMBER(4),

A_id NUMBER(4));



INSERT INTO STUDENT VALUES

(101, 'TASFIA',1,2001);

INSERT INTO STUDENT VALUES

(102, 'TAWAZ',2,2002);

INSERT INTO STUDENT VALUES

(103, 'ATHOY',3,2003);

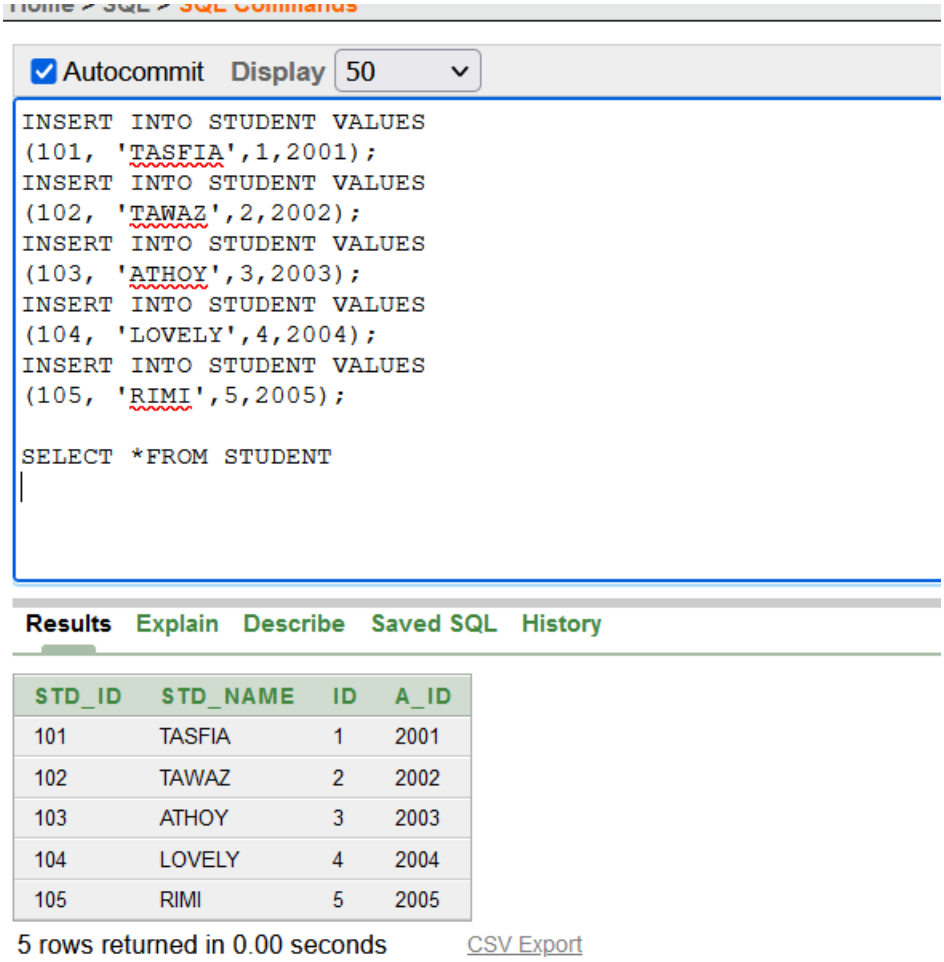
INSERT INTO STUDENT VALUES

(104, 'LOVELY',4,2004);

INSERT INTO STUDENT VALUES

(105, 'RIMI',5,2005);

SELECT *FROM STUDENT

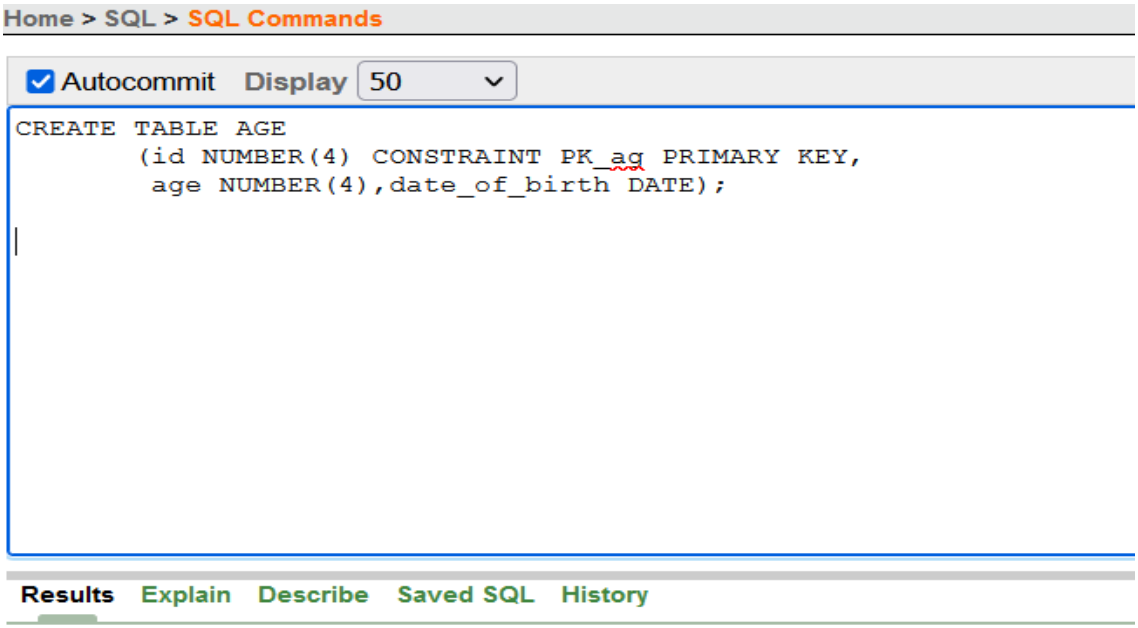


AGE

CREATE TABLE AGE

(id NUMBER(4) CONSTRAINT PK_ag PRIMARY KEY,

age NUMBER(4),date_of_birth DATE);




```
Table created.
```

0.02 seconds

INSERT INTO AGE VALUES

(1, 21, to_date('01-05-2000','dd-mm-yyyy'));

INSERT INTO AGE VALUES

(2, 23, to_date('08-08-1999','dd-mm-yyyy'));

INSERT INTO AGE VALUES

(3, 25, to_date('12-01-1997','dd-mm-yyyy'));

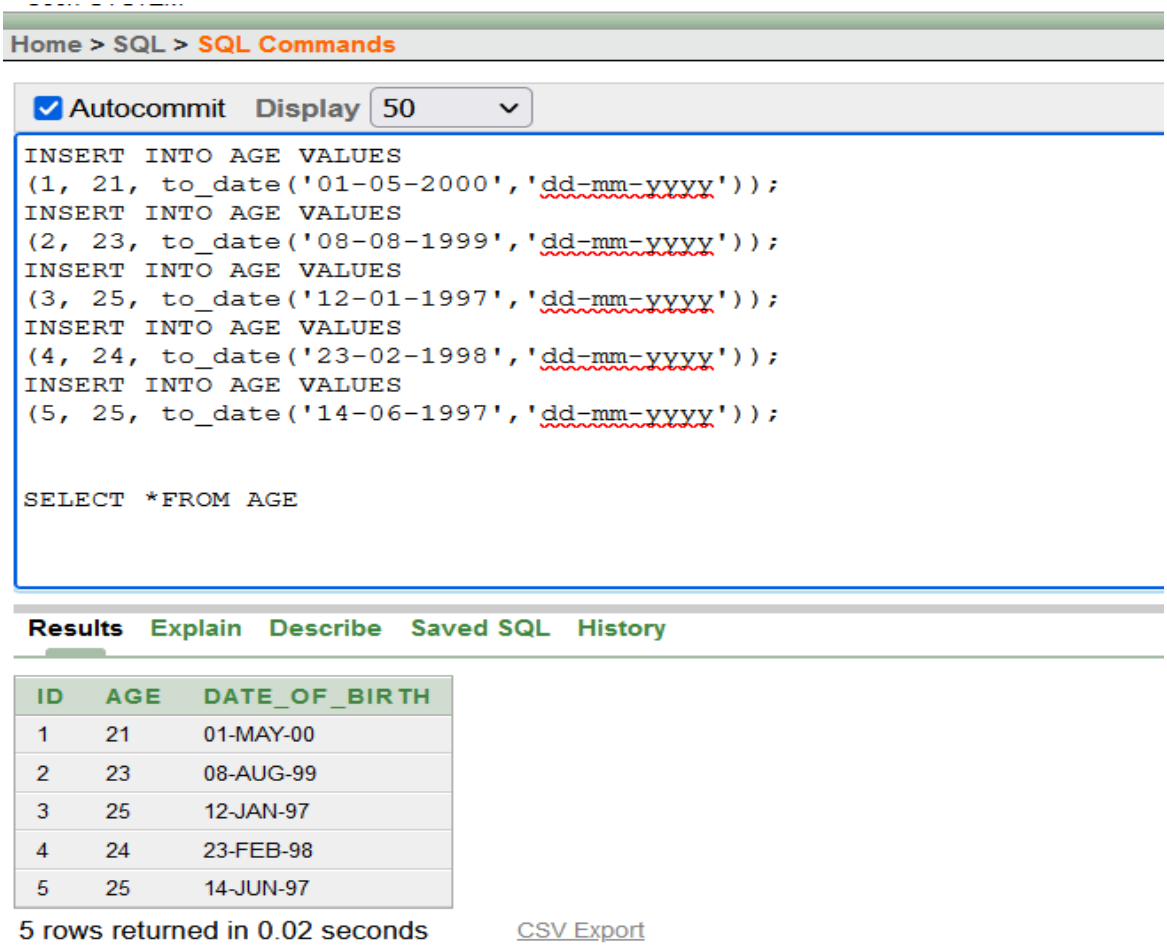
INSERT INTO AGE VALUES

(4, 24, to_date('23-02-1998','dd-mm-yyyy'));

INSERT INTO AGE VALUES

(5, 25, to_date('14-06-1997','dd-mm-yyyy'));

SELECT *FROM AGE



FACULTY

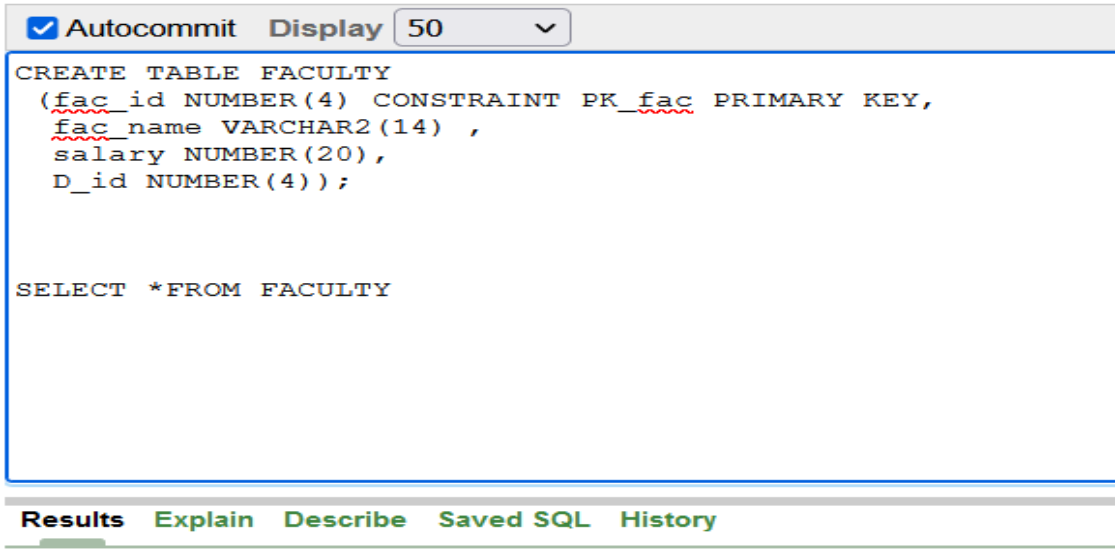
CREATE TABLE FACULTY

(fac_id NUMBER(4) CONSTRAINT PK_fac PRIMARY KEY,

fac_name VARCHAR2(14) ,

salary NUMBER(20),

D_id NUMBER(4));




```
Table created.
```

0.00 seconds

INSERT INTO FACULTY VALUES

(111, 'NAZIA ALFAZ', 40000, 10);

INSERT INTO FACULTY VALUES

(222, 'RIFAT MAHMUD', 45000, 20);

INSERT INTO FACULTY VALUES

(333, 'JUENA NOWSHIN', 50000, 30);

INSERT INTO FACULTY VALUES

(444, 'KAMRUN NAHAR', 70000, 40);

INSERT INTO FACULTY VALUES

(555, 'BITHI PAUL', 35000, 50);

SELECT *FROM FACULTY

Autocommit Display 50

```
INSERT INTO FACULTY VALUES
(111, 'NAZIA ALFAZ', 40000, 10);
INSERT INTO FACULTY VALUES
(222, 'RIFAT MAHMUD', 45000, 20);
INSERT INTO FACULTY VALUES
(333, 'JUENA NOWSHIN', 50000, 30);
INSERT INTO FACULTY VALUES
(444, 'KAMRUN NAHAR', 70000, 40);
INSERT INTO FACULTY VALUES
(555, 'BITHI PAUL', 35000, 50);


SELECT *FROM FACULTY
```

Results Explain Describe Saved SQL History

FAC_ID	FAC_NAME	SALARY	D_ID
111	NAZIA ALFAZ	40000	10
222	RIFAT MAHMUD	45000	20
333	JUENA NOWSHIN	50000	30
444	KAMRUN NAHAR	70000	40
555	BITHI PAUL	35000	50

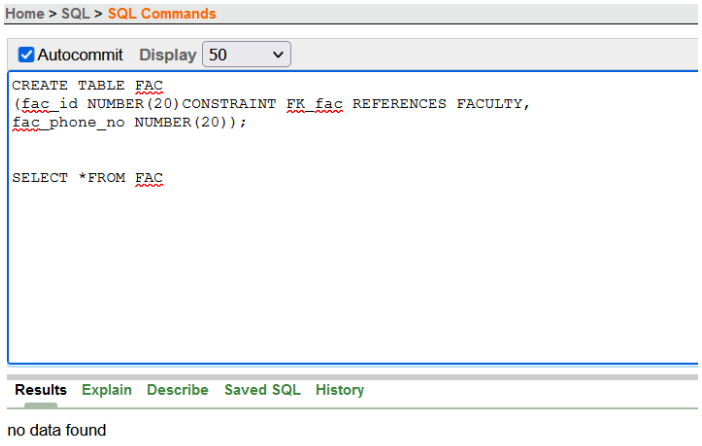
5 rows returned in 0.00 seconds CSV Export

FAC

CREATE TABLE FAC

(fac_id NUMBER(20)CONSTRAINT FK_fac REFERENCES FACULTY,

fac_phone_no NUMBER(20));



INSERT INTO FAC VALUES

(111, 01731122975);

INSERT INTO FAC VALUES

(222, 01843555335);

INSERT INTO FAC VALUES

(333, 01714345487);

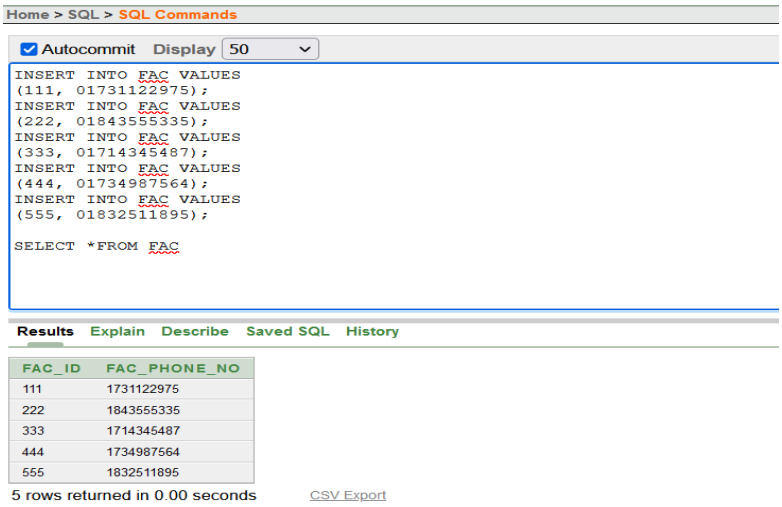
INSERT INTO FAC VALUES

(444, 01734987564);

INSERT INTO FAC VALUES

(555, 01832511895);

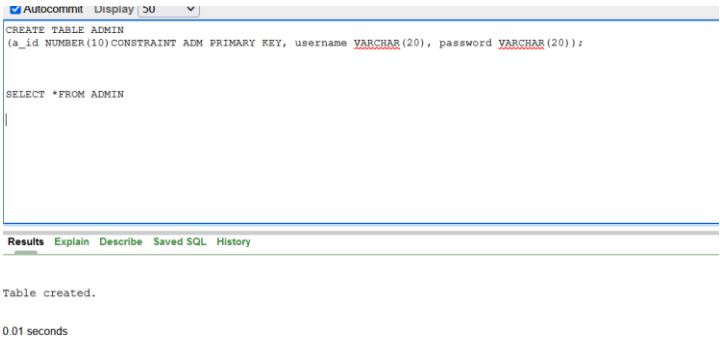
SELECT *FROM FAC



ADMIN

CREATE TABLE ADMIN

(a_id NUMBER(10)CONSTRAINT ADM PRIMARY KEY, username VARCHAR(20), password VARCHAR(20));



INSERT INTO ADMIN VALUES

(30001, 'BLUE', 'SYSTEM');

INSERT INTO ADMIN VALUES

(30002, 'BLACK', 'SCOTT');

INSERT INTO ADMIN VALUES

(30003, 'OFFWHITE', 'TIGER');

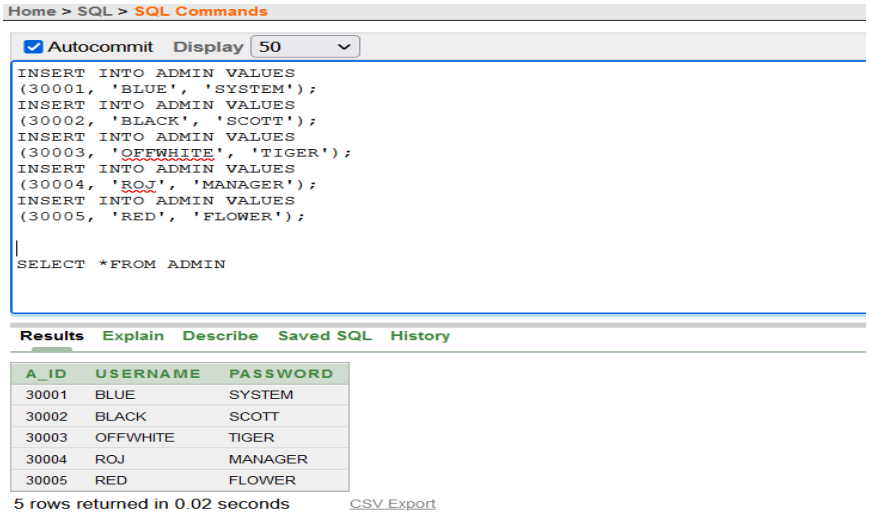
INSERT INTO ADMIN VALUES

(30004, 'ROJ', 'MANAGER');

INSERT INTO ADMIN VALUES

(30005, 'RED', 'FLOWER');

SELECT *FROM ADMIN

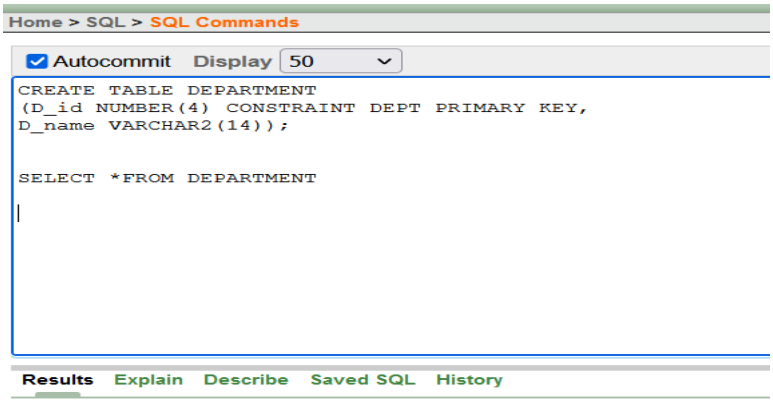


DEPARTMENT

CREATE TABLE DEPARTMENT

(D_id NUMBER(4) CONSTRAINT DEPT PRIMARY KEY,

D_name VARCHAR2(14));



INSERT INTO DEPARTMENT VALUES

(10, 'EEE');

INSERT INTO DEPARTMENT VALUES

(20, 'CSE');

INSERT INTO DEPARTMENT VALUES

(30, 'BBA');

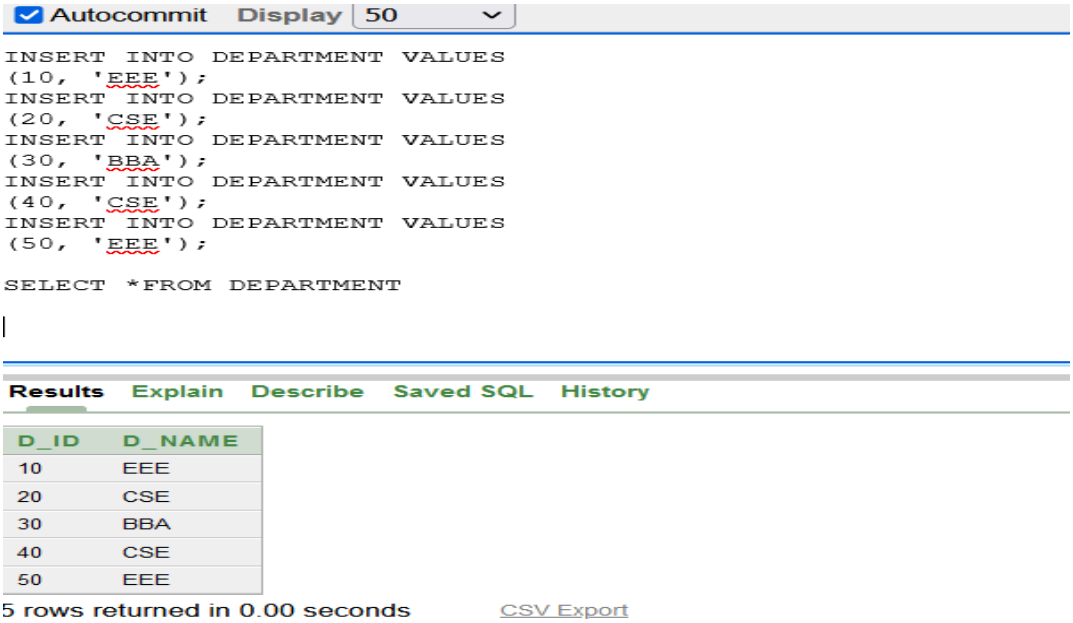
INSERT INTO DEPARTMENT VALUES

(40, 'CSE');

INSERT INTO DEPARTMENT VALUES

(50, 'EEE');

SELECT *FROM DEPARTMENT

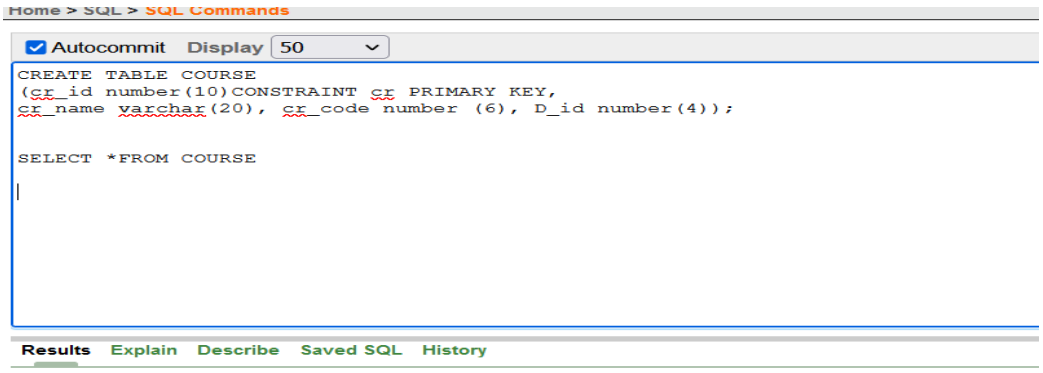


COURSE

CREATE TABLE COURSE

(cr_id number(10)CONSTRAINT cr PRIMARY KEY,

cr_name varchar(20), cr_code number (6), D_id number(4));



INSERT INTO COURSE VALUES

(5501, 'C++', 005823, 10);

INSERT INTO COURSE VALUES

(5502, 'DATABASE', 041519, 20);

INSERT INTO COURSE VALUES

(5503, 'JAVA', 011893, 30);

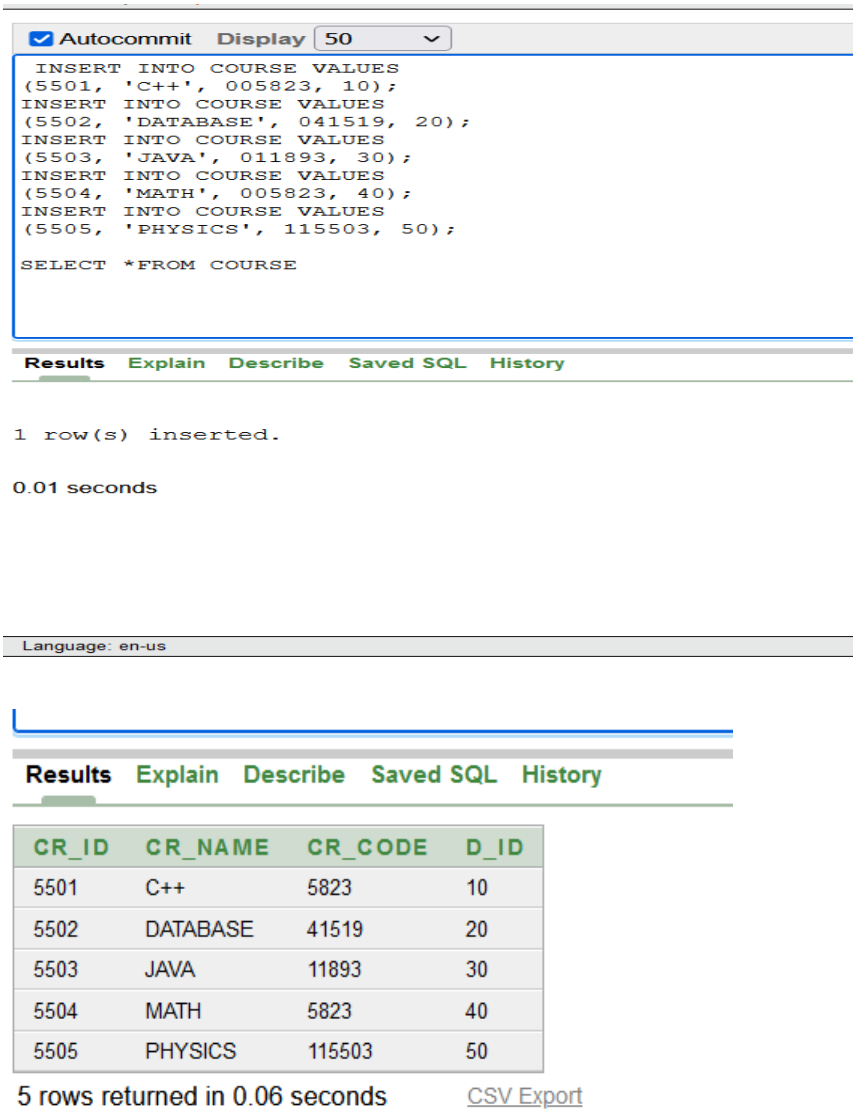
INSERT INTO COURSE VALUES

(5504, 'MATH', 005823, 40);

INSERT INTO COURSE VALUES

(5505, 'PHYSICS', 115503, 50);

SELECT *FROM COURSE



Results Explain Describe Saved SQL History

CR_ID	CR_NAME	CR_CODE	D_ID
5501	C++	5823	10
5502	DATABASE	41519	20
5503	JAVA	11893	30
5504	MATH	5823	40
5505	PHYSICS	115503	50

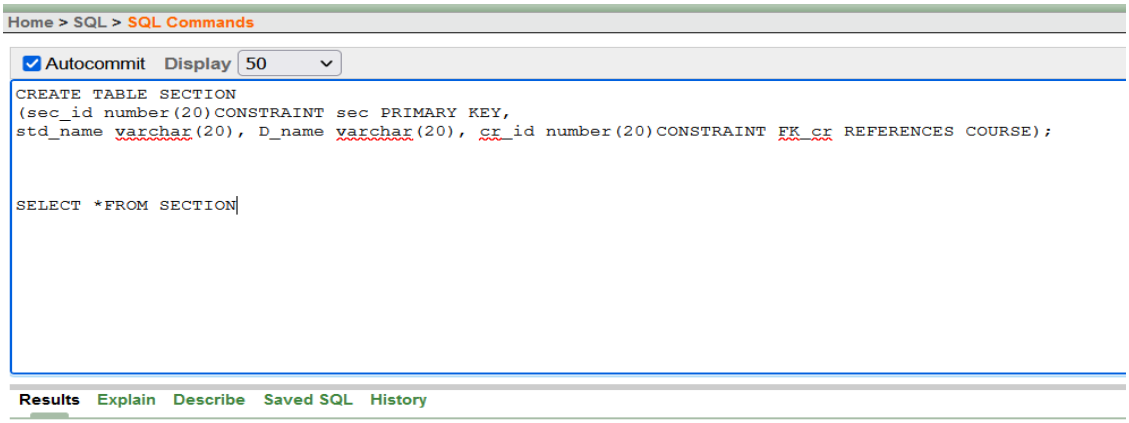
5 rows returned in 0.06 seconds CSV Export

SECTION

CREATE TABLE SECTION

(sec_id number(20)CONSTRAINT sec PRIMARY KEY,

std_name varchar(20), D_name varchar(20), cr_id number(20)CONSTRAINT FK_cr REFERENCES COURSE);




```
Table created.

0.03 seconds
```

INSERT INTO Section VALUES

(201, 'TASFIA', 'EEE', 5501);

INSERT INTO Section VALUES

(202, 'TAWAZ', 'CSE', 5502);

INSERT INTO Section VALUES

(203, 'ATHOY', 'BBA', 5503);

INSERT INTO Section VALUES

(204, 'LOVELY', 'CSE', 5504);

INSERT INTO Section VALUES

(205, 'RIMI', 'EEE', 5505);

SELECT *FROM SECTION

Autocommit Display 50

```
INSERT INTO Section VALUES
(201, 'TASFIA', 'EEE', 5501);
INSERT INTO Section VALUES
(202, 'TAWAZ', 'CSE', 5502);
INSERT INTO Section VALUES
(203, 'ATHOY', 'BBA', 5503);
INSERT INTO Section VALUES
(204, 'LOVELY', 'CSE', 5504);
INSERT INTO Section VALUES
(205, 'RIMI', 'EEE', 5505);

SELECT *FROM SECTION
```

Results Explain Describe Saved SQL History

SEC_ID	STD_NAME	D_NAME	CR_ID
201	TASFIA	EEE	5501
202	TAWAZ	CSE	5502
203	ATHOY	BBA	5503
204	LOVELY	CSE	5504
205	RIMI	EEE	5505

5 rows returned in 0.00 seconds CSV Export

ATTENDENCE

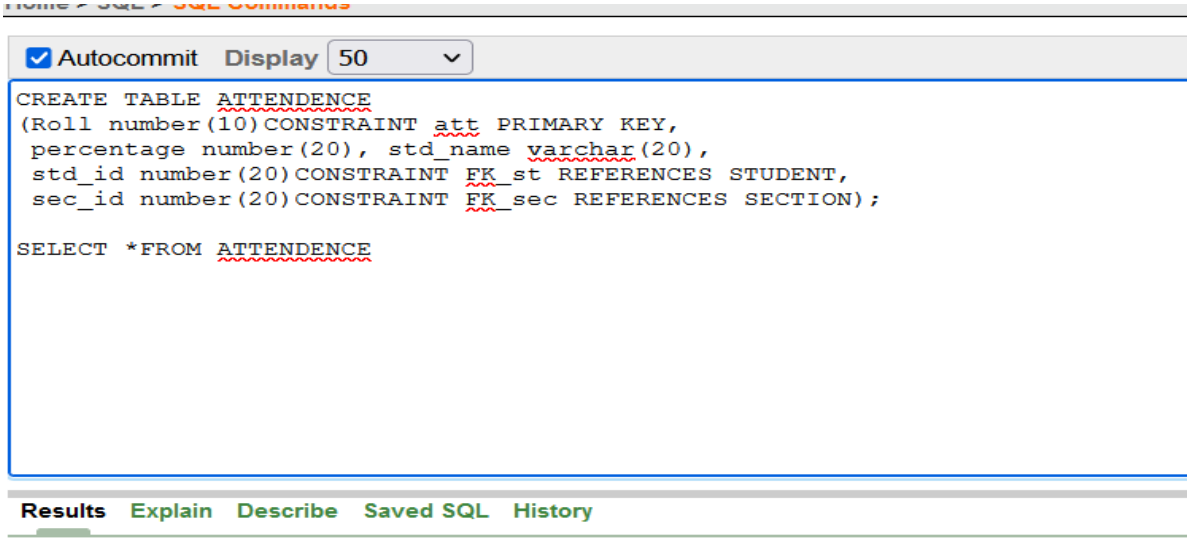
CREATE TABLE ATTENDENCE

(Roll number(10)CONSTRAINT att PRIMARY KEY,

percentage number(20), std_name varchar(20),

std_id number(20)CONSTRAINT FK_st REFERENCES STUDENT,

sec_id number(20)CONSTRAINT FK_sec REFERENCES SECTION);




```
Table created.
```

0.01 seconds

INSERT INTO ATTENDENCE VALUES

(01, 98, 'TASFIA', 101, 201);

INSERT INTO ATTENDENCE VALUES

(02, 88, 'TAWAZ', 102, 202);

INSERT INTO ATTENDENCE VALUES

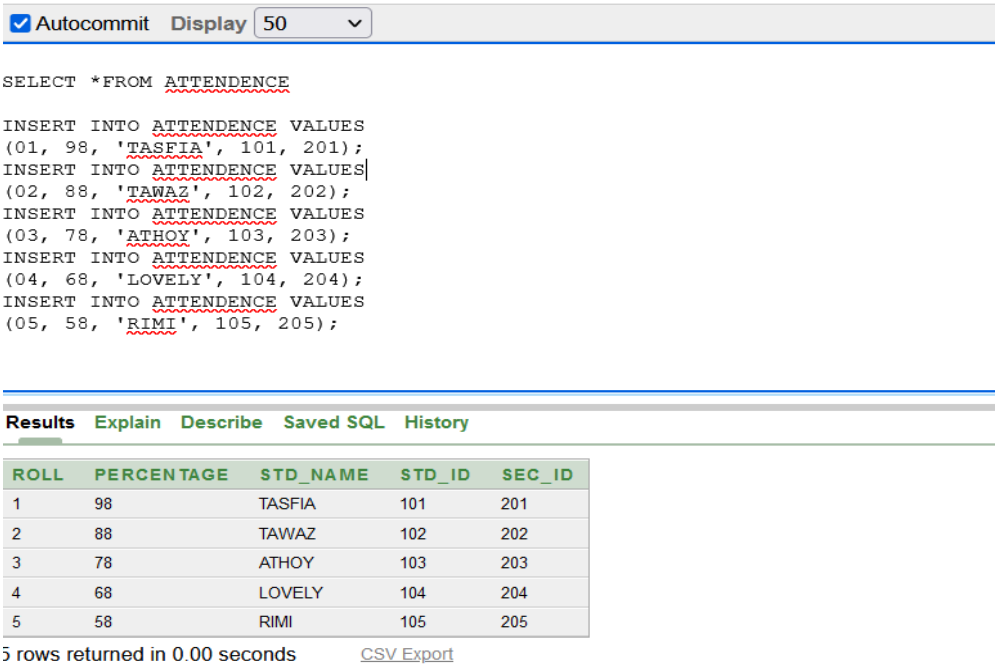
(03, 78, 'ATHOY', 103, 203);

INSERT INTO ATTENDENCE VALUES

(04, 68, 'LOVELY', 104, 204);

INSERT INTO ATTENDENCE VALUES

(05, 58, 'RIMI', 105, 205);

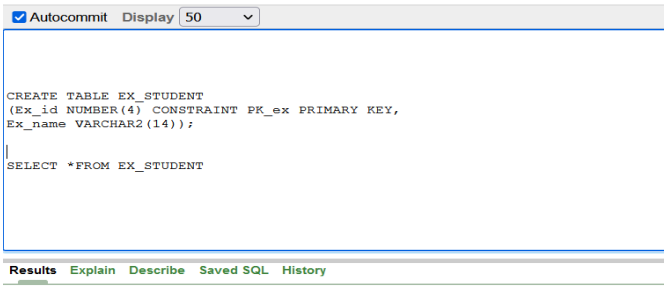


EX-STUDENT

CREATE TABLE EX_STUDENT

(Ex_id NUMBER(4) CONSTRAINT PK_ex PRIMARY KEY,

Ex_name VARCHAR2(14));



INSERT INTO EX_STUDENT VALUES

(91, 'MAISHA');

INSERT INTO EX_STUDENT VALUES

(92, 'ROJ');

```
INSERT INTO EX_STUDENT VALUES
(91, 'MAISHA');

INSERT INTO EX_STUDENT VALUES
(92, 'ROJ');


SELECT *FROM EX_STUDENT
```

Results Explain Describe Saved SQL History

EX_ID	EX_NAME
91	MAISHA
92	ROJ

2 rows returned in 0.00 seconds CSV Export

CURRENT-STUDENT

CREATE TABLE CURRENT_STUDENT

(cur_id NUMBER(4) CONSTRAINT PK_cur PRIMARY KEY,

cur_name VARCHAR2(14));

```
CREATE TABLE CURRENT_STUDENT
(cur_id NUMBER(4) CONSTRAINT PK_cur PRIMARY KEY,
cur_name VARCHAR2(14));


SELECT *FROM CURRENT_STUDENT
```

Results Explain Describe Saved SQL History

```
Table created.
```

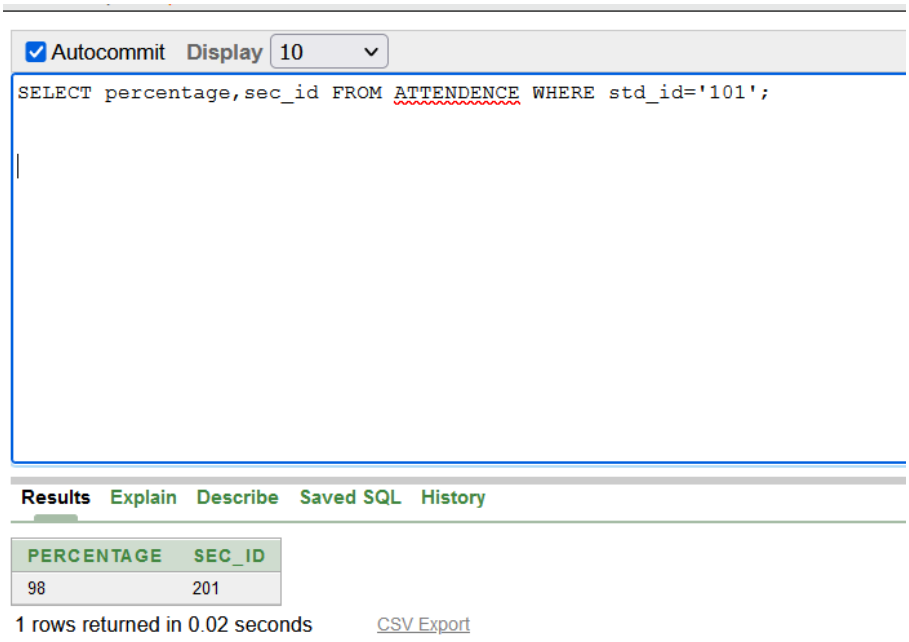
0.01 seconds

QUERY WRITING

SIMPLE QUERY

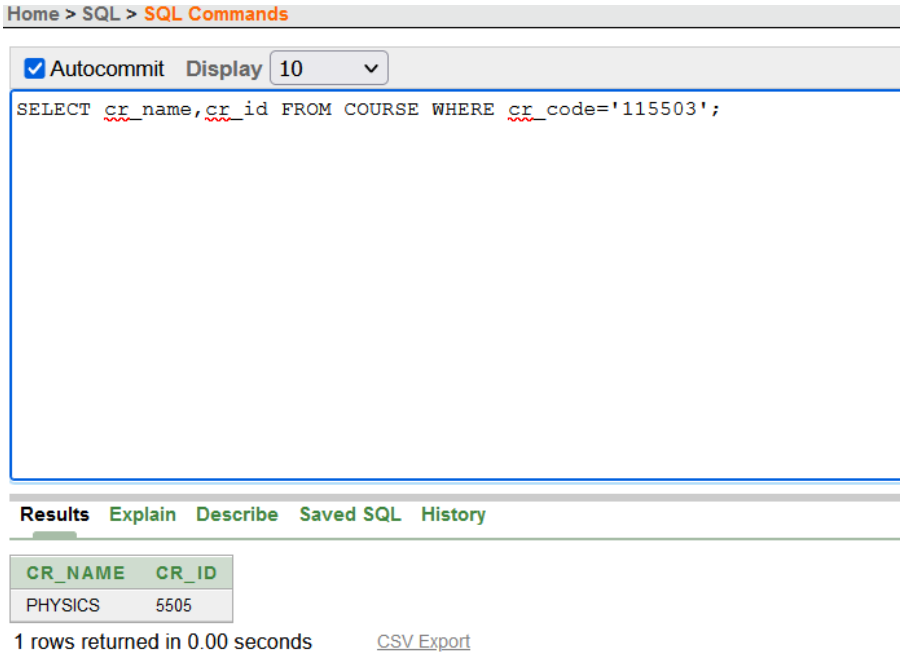
1.Display the percentage and sec_id for std_id ‘101’

SELECT percentage,sec_id FROM ATTENDENCE WHERE std_id='101';



2. Display cr_name and cr_id from course where cr_code is ‘115503’

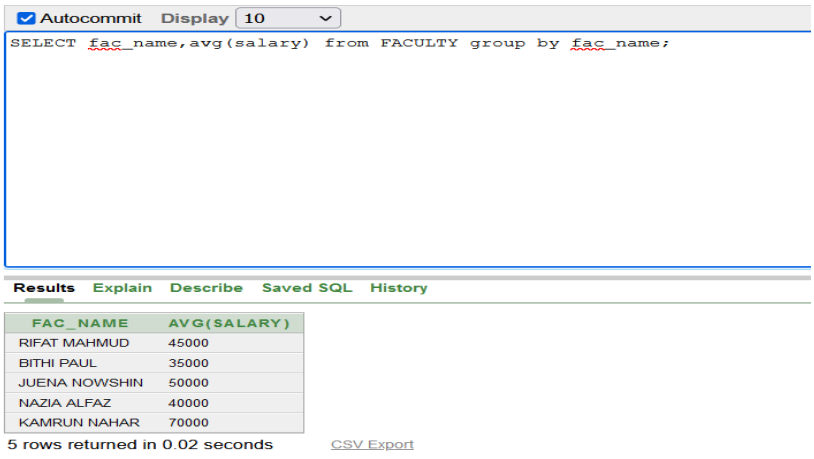
SELECT cr_name,cr_id FROM COURSE WHERE cr_code='115503';



SINGLE ROW FUNCTION

1.Display Faculty name and average salary and group by names

SELECT fac_name,avg(salary) from FACULTY group by fac_name;



MULTIPLE RAW FUNCTION

1. Display max(salary) in faculty table

SELECT MAX(SALARY)FROM FACULTY

```
SELECT MAX(SALARY) FROM FACULTY
```

Results Explain Describe Saved SQL History

MAX(SALARY)
70000

1 rows returned in 0.00 seconds CSV Export

SINGLE ROW SUB QUERY

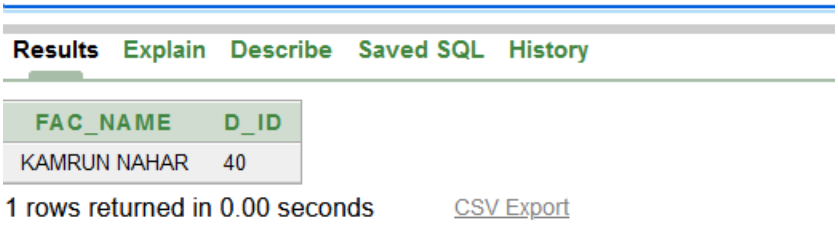
1. Write a query to display fac_name, d_id and max salary of those faculties

SELECT fac_name, d_id

FROM Faculty

WHERE salary=(select max(salary) from Faculty);

```
SELECT fac_name, d_id
FROM Faculty
WHERE salary=(select max(salary) from Faculty);
```



Results Explain Describe Saved SQL History

FAC_NAME	D_ID
KAMRUN NAHAR	40

1 rows returned in 0.00 seconds CSV Export

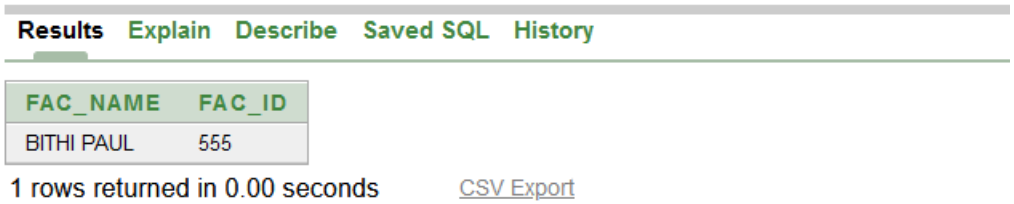
2.Write a query to display fac_name, d_id and min salary of those faculties

SELECT fac_name, fac_id

FROM Faculty

WHERE salary=(select min(salary) from Faculty);

```
SELECT fac_name, fac_id
FROM Faculty
WHERE salary=(select min(salary) from Faculty);
```



Results Explain Describe Saved SQL History

FAC_NAME	FAC_ID
BITHI PAUL	555

1 rows returned in 0.00 seconds CSV Export

MULTI ROW SUB QUERY

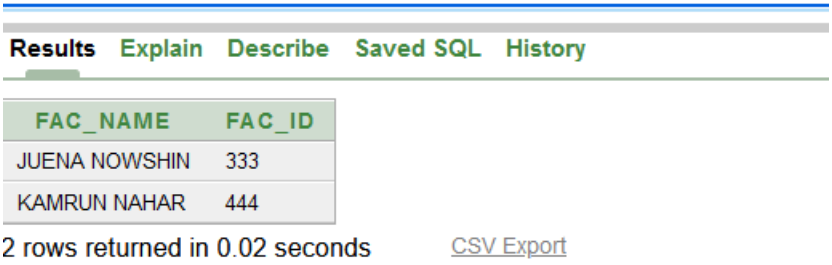
1. Write a query to display fac_name, fac_id and find the average salary of all those faculties

SELECT fac_name, fac_id

FROM Faculty

WHERE salary>ALL(select avg(salary) from Faculty);

```
SELECT fac_name, fac_id
FROM Faculty
WHERE salary>ALL(select avg(salary) from Faculty);
```



Results Explain Describe Saved SQL History

FAC_NAME	FAC_ID
JUENA NOWSHIN	333
KAMRUN NAHAR	444

2 rows returned in 0.02 seconds CSV Export

COMPLEX SUB QUERY

1. Write a query to display d_id and fac_name='NAZIA ALFAZ' in faculty

SELECT *

FROM FACULTY

WHERE d_id = (SELECT d_id

FROM FACULTY

WHERE fac_name='NAZIA ALFAZ');

```
SELECT *
FROM FACULTY
WHERE d_id = (SELECT d_id
              FROM FACULTY
              WHERE fac_name='NAZIA ALFAZ');
```

Results Explain Describe Saved SQL History

FAC_ID	FAC_NAME	SALARY	D_ID
111	NAZIA ALFAZ	40000	10

1 rows returned in 0.02 seconds CSV Export

2.Write a query to display d_id and find cr_name='JAVA'course in faculty

SELECT *

FROM COURSE

WHERE d_id <ANY (SELECT d_id

FROM FACULTY

WHERE cr_name='JAVA');

```
SELECT *
FROM COURSE
WHERE d_id <ANY (SELECT d_id
                 FROM FACULTY
                 WHERE cr_name='JAVA');
```

Results Explain Describe Saved SQL History

CR_ID	CR_NAME	CR_CODE	D_ID
5503	JAVA	11893	30

1 rows returned in 0.02 seconds CSV Export

EQUI-JOIN

1.Joining the fac_name from Faculty table and D_name from Department table using equi-join as Faculty and Department has direct relation.

SELECT w.fac_name, c.d_name

FROM FACULTY w , Department c

WHERE w.d_id = c.d_id;

```
SELECT w.fac_name, c.d_name
FROM  FACULTY w , Department c
WHERE w.d_id = c.d_id;
```

Results | Explain | Describe | Saved SQL | History

FAC_NAME	D_NAME
NAZIA ALFAZ	EEE
RIFAT MAHMUD	CSE
JUENA NOWSHIN	BBA
KAMRUN NAHAR	CSE
BITHI PAUL	EEE

5 rows returned in 0.01 seconds CSV Export

LEFT OUTER JOIN

1. Joining the fac_name,salary from Faculty table and D_name from Department table using left outer join as Faculty and Department has direct relation.

SELECT w.fac_name,w.salary,c.d_name

FROM FACULTY w , Department c

WHERE w.d_id = c.d_id(+);

```
SELECT w.fac_name,w.salary,c.d_name
FROM  FACULTY w , Department c
WHERE w.d_id = c.d_id(+);
```

Results Explain Describe Saved SQL History

FAC_NAME	SALARY	D_NAME
NAZIA ALFAZ	40000	EEE
RIFAT MAHMUD	45000	CSE
JUENA NOWSHIN	50000	BBA
KAMRUN NAHAR	70000	CSE
BITHI PAUL	35000	EEE

5 rows returned in 0.00 seconds CSV Export

RIGHT OUTER JOIN

1. Joining the fac_name,fac_id from Faculty table and cr_name from course table using right outer join as Faculty and course has direct relation.

SELECT w.fac_name,w.fac_id,c.cr_name

FROM FACULTY w ,COURSE c

WHERE w.d_id(+) = c.d_id;

```
SELECT w.fac_name,w.fac_id,c.cr_name
FROM  FACULTY w ,COURSE c
WHERE w.d_id(+) = c.d_id;
```

Results Explain Describe Saved SQL History

FAC_NAME	FAC_ID	CR_NAME
NAZIA ALFAZ	111	C++
RIFAT MAHMUD	222	DATABASE
JUENA NOWSHIN	333	JAVA
KAMRUN NAHAR	444	MATH
BITHI PAUL	555	PHYSICS

5 rows returned in 0.00 seconds CSV Export

FULL OUTER JOIN

SELECT *

FROM FACULTY w , COURSE c

WHERE w.d_id = c.d_id(+);

```
SELECT *
FROM  FACULTY w , COURSE c
WHERE w.d_id = c.d_id(+);
```

Results Explain Describe Saved SQL History

FAC_ID	FAC_NAME	SALARY	D_ID	CR_ID	CR_NAME	CR_CODE	D_ID
111	NAZIA ALFAZ	40000	10	5501	C++	5823	10
222	RIFAT MAHMUD	45000	20	5502	DATABASE	41519	20
333	JUENA NOWSHIN	50000	30	5503	JAVA	11893	30
444	KAMRUN NAHAR	70000	40	5504	MATH	5823	40
555	BITHI PAUL	35000	50	5505	PHYSICS	115503	50

5 rows returned in 0.00 seconds CSV Export

NON EQUI JOIN

select faculty.fac_id,faculty.fac_name,department.d_name,department.d_id from faculty,department where department.d_id<faculty.d_id;

```
select faculty.fac_id,faculty.fac_name,department.d_name,department.d_id from
faculty,department where department.d_id<faculty.d_id;
```

Results Explain Describe Saved SQL History

FAC_ID	FAC_NAME	D_NAME	D_ID
222	RIFAT MAHMUD	EEE	10
333	JUENA NOWSHIN	EEE	10
333	JUENA NOWSHIN	CSE	20
444	KAMRUN NAHAR	EEE	10
444	KAMRUN NAHAR	CSE	20
444	KAMRUN NAHAR	BBA	30
555	BITHI PAUL	EEE	10
555	BITHI PAUL	CSE	20
555	BITHI PAUL	BBA	30
555	BITHI PAUL	CSE	40

10 rows returned in 0.00 seconds CSV Export

SELF OUTER JOIN

SELECT p.fac_name||' is the faculty of the department '|| i.d_name

FROM FACULTY p ,Department i

WHERE p.d_id(+) = i.d_id;

```
SELECT p.fac_name||' is the faculty of the department '|| i.d_name
FROM FACULTY p ,Department i
WHERE p.d_id(+) = i.d_id;
```

Results Explain Describe Saved SQL History

P.FAC_NAME||'ISTHEFACULTYOFTHEDEPARTMENT'||I.D_NAME
NAZIA ALFAZ is the faculty of the department EEE
RIFAT MAHMUD is the faculty of the department CSE
JUENA NOWSHIN is the faculty of the department BBA
KAMRUN NAHAR is the faculty of the department CSE
BITHI PAUL is the faculty of the department EEE

5 rows returned in 0.02 seconds CSV Export

VIEW

1.create a view WORKLOAD , to identidy date of birth whom age is 23

CREATE VIEW WORKLOAD

As select id,date_of_birth

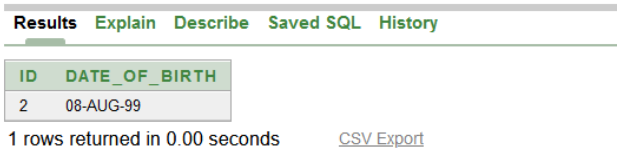
FROM AGE

WHERE AGE ='23';

select * from WORKLOAD

```
CREATE VIEW WORKLOAD
As select id,date_of_birth
FROM AGE
WHERE AGE ='23';

select * from WORKLOAD
```



ID	DATE_OF_BIRTH
2	08-AUG-99

1 rows returned in 0.00 seconds CSV Export

SEQUENCE

1. Create a sequence to auto generate the department id by the system starting from 60 and no cycle

CREATE SEQUENCE DEPT_SEQ

MINVALUE 10

MAXVALUE 100

START WITH 60

NOCACHE

NOCYCLE

INCREMENT BY 10;

```
CREATE SEQUENCE DEPT_SEQ
    MINVALUE 10
    MAXVALUE 100
    START WITH 60
    NOCACHE
    NOCYCLE
    INCREMENT BY 10;
```

Results Explain Describe Saved SQL History

```
Sequence created.
```

0.02 seconds

INSERT INTO DEPARTMENT VALUES (DEPT_SEQ.NEXTVAL, 'EEE');

INSERT INTO DEPARTMENT VALUES (DEPT_SEQ.NEXTVAL, 'BBA');

SELECT * FROM DEPARTMENT;

```
INSERT INTO DEPARTMENT VALUES (DEPT_SEQ.NEXTVAL, 'EEE');
INSERT INTO DEPARTMENT VALUES (DEPT_SEQ.NEXTVAL, 'BBA');
SELECT * FROM DEPARTMENT;
```

Results Explain Describe Saved SQL History

D_ID	D_NAME
10	EEE
20	CSE
30	BBA
40	CSE
50	EEE
60	EEE
70	BBA

7 rows returned in 0.00 seconds CSV Export