University Management

DBMS-Project

Jayanth Kumar Goud

ABSTRACT:

A university is a high-level educational institution in which students study for degrees and academic research is done. Here, we'll create a database for the university as we have data for various entities such as Student, faculty, course and many more. Each student has a particular name and same goes with the faculty members. A student is a part of only one department while the department may have many number of students or have no one because it's optional. There are various departments. Each department has its own Faculty members teaching various Subjects. Students enroll in various subjects taught by the respective faculty. Students are graded based on their performance in their respective Examinations. We have created Student Log relation to keep track of updates to the student database. We have the relation Accounts between the student and the university. The students pay fee as per their course and scholarship.

REQUIREMENT ANALYSIS:

EXISTING SYSTEM

The system starts with registration of new staff and students. When the subjects are to be allocated to the faculty, the Head of the Department should enter everything in the Excel sheets. Then the staff enters corresponding subject's attendance and marks of a student then those must also be entered in the Excel sheets and validations are to be done by the user itself. So there will be a lot of work to be done and must be more conscious during the entrance of details. So, more risk is involved.

PROBLEMS IN THE EXISTING SYSTEM:

Storing and accessing the data in the form of Excel sheets and account books is a tedious work. It requires a lot of laborious work. It may often yield undesired results. Maintaining these records as piles may turn out to be a costlier task than any other of the colleges and institutions.

Part-I- Data Modeling Design

a) Data Requirements –

i.Entities

University

Department

Faculty

Subject

Teaches

Student

Studlog

Exam

Enrolled

Grades

Accounts

ii.Relationships

Faculty work in departments

Students enroll in subject

Faculty teach student

Students take exam and get grades

Students pay fees to the university

iii.Attributes

I. UNIVERSITY(BRANCH_ID, BRANCH_NAME, BRANCH_ADDRESS)

II. DEPARTMENT(DEP_I, DEP_NAME)

III. FACULTY(FAC_I, FAC_NAME, ROOM_NO, DEP_ID)

IV. SUBJECT(SUB_ID, SUB_NAME, FAC_ID, DEP_ID)

V. TEACHES(FAC_ID, , SUB_ID)

VI. STUDENT(STUD_ID, STUD_NAME, , GENDER, CGPA, AGE)

VII. STUDLOG(STUD_ID, STUD_NAME, , GENDER, CGPA, AGE)

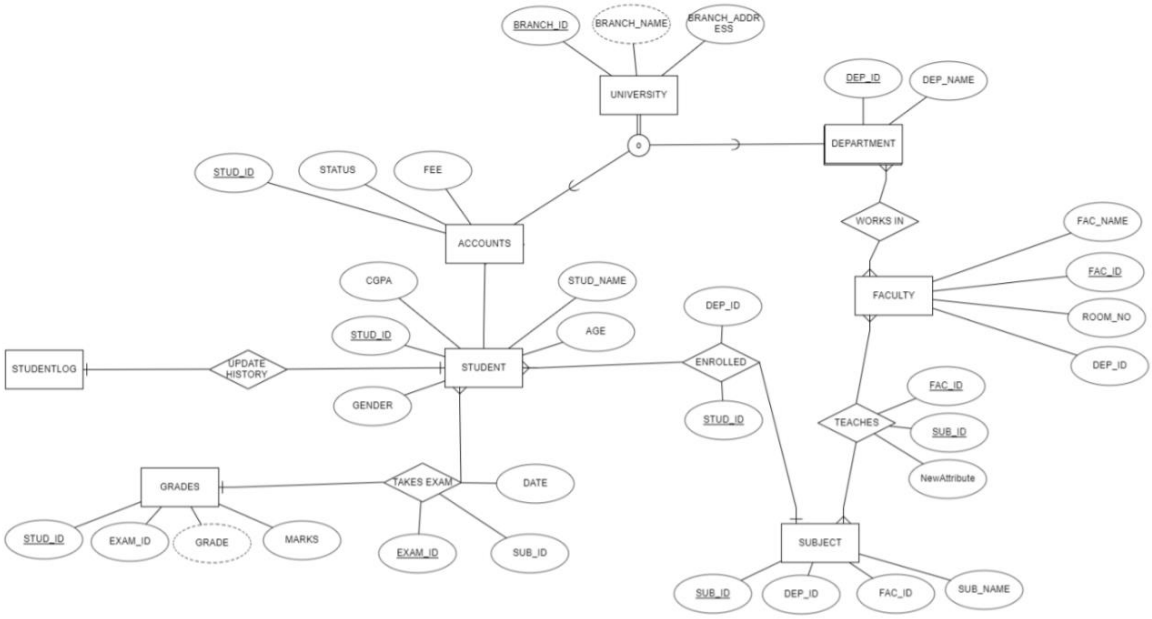
VIII. EXAM(EXAM_ID, SUB_ID, DATE)

IX. ENROLLED(STUDENT_ID VARCHAR, DEP_ID)

X. GRADES(STUD_ID, EXAM_ID, MARKS, GRADE)

XI. ACCOUNTS(STUD_ID, FEE, STATUS) b)ER Diagram

b)ER Diagram



Part-II-Relational Database Schema Development

Conceptual Schema

1) UNIVERSITY(BRANCH_ID VARCHAR, BRANCH_NAME VARCHAR, BRANCH_ADDRESS VARCHAR)

2) DEPARTMENT(DEP_ID VARCHAR, DEP_NAME VARCHAR)

3) FACULTY(FAC_ID VARCHAR, FAC_NAME VARCHAR, ROOM_NO INTEGER, DEP_ID VARCHAR)

4) SUBJECT(SUB_ID VARCHAR, SUB_NAME VARCHAR, FAC_ID VARCHAR, DEP_ID VARCHAR)

5) TEACHES(FAC_ID, VARCHAR, SUB_ID VARCHAR)

6) STUDENT(STUD_ID VARCHAR, STUD_NAME, VARCHAR, GENDER VARCHAR, CGPA REAL, AGE INTEGER)

7) STUDLOG(STUD_ID VARCHAR, STUD_NAME, VARCHAR, GENDER VARCHAR, CGPA REAL, AGE INTEGER)

8) EXAM(EXAM_ID VARCHAR, SUB_ID VARCHAR, DATE DATE)

9) ENROLLED(STUDENT_ID VARCHAR, DEP_ID VARCHAR)

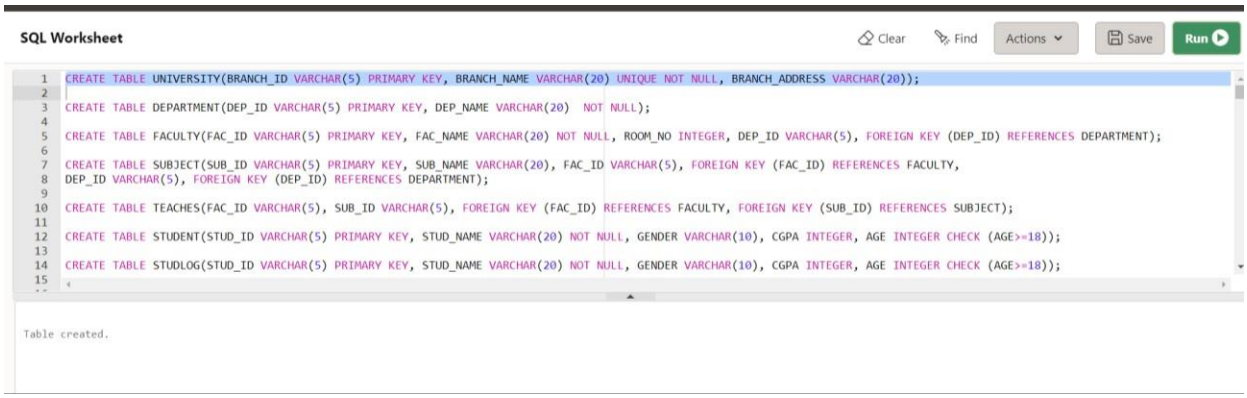
10) GRADES(STUD_ID VARCHAR, EXAM_ID VARCHAR, MARKS INTEGER, GRADE VARCHAR)

11) ACCOUNTS(STUD_ID VARCHAR, FEE INTEGER, STATUS VARCHAR)

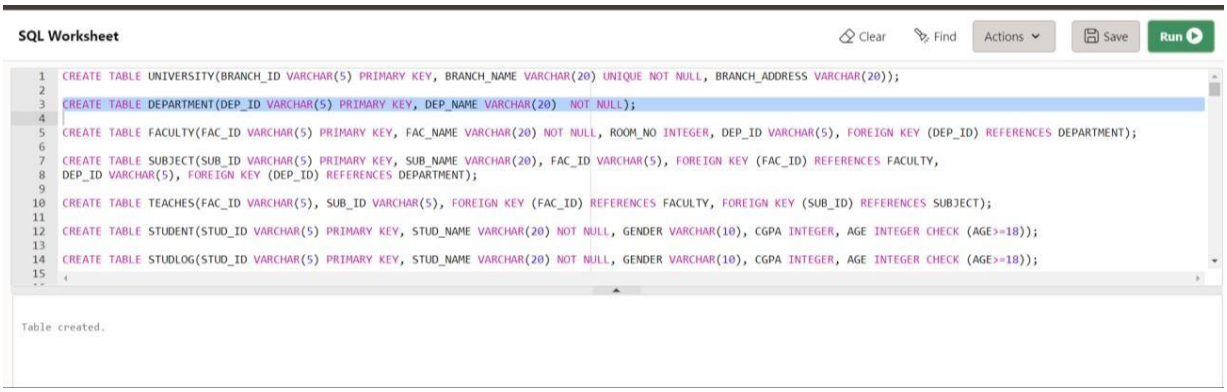
Part-III- Relational Database Implementation

- **Tables Creation**

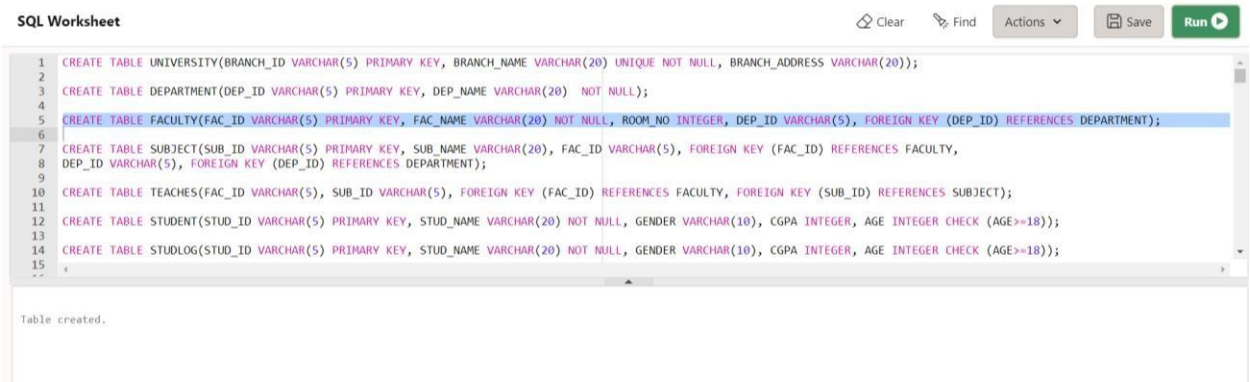
CREATE TABLE UNIVERSITY(BRANCH_ID VARCHAR(5) PRIMARY KEY, BRANCH_NAME VARCHAR(20) UNIQUE NOT NULL, BRANCH_ADDRESS VARCHAR(20));



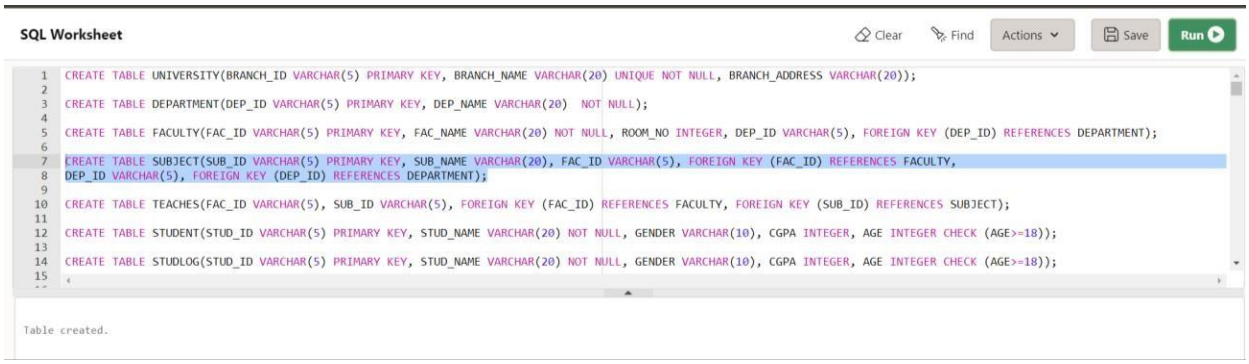
CREATE TABLE DEPARTMENT(DEP_ID VARCHAR(5) PRIMARY KEY, DEP_NAME VARCHAR(20) NOT NULL);



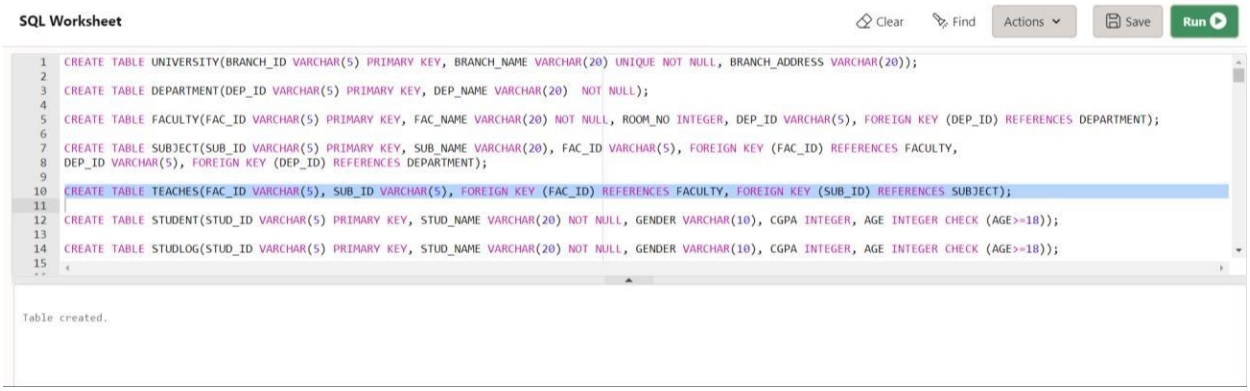
CREATE TABLE FACULTY(FAC_ID VARCHAR(5) PRIMARY KEY, FAC_NAME VARCHAR(20) NOT NULL, ROOM_NO INTEGER, DEP_ID VARCHAR(5), FOREIGN KEY (DEP_ID) REFERENCES DEPARTMENT);



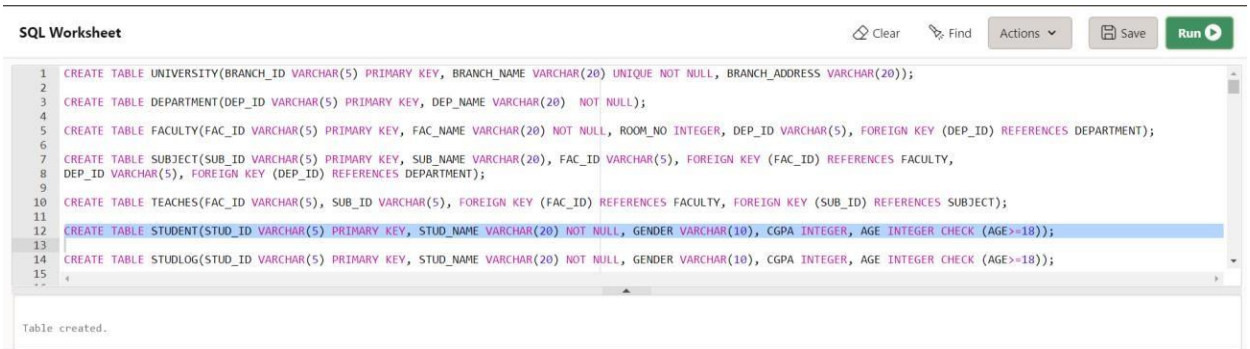
CREATE TABLE SUBJECT(SUB_ID VARCHAR(5) PRIMARY KEY, SUB_NAME VARCHAR(20), FAC_ID VARCHAR(5), FOREIGN KEY (FAC_ID) REFERENCES FACULTY,
DEP_ID VARCHAR(5), FOREIGN KEY (DEP_ID) REFERENCES DEPARTMENT);



CREATE TABLE TEACHES(FAC_ID VARCHAR(5), SUB_ID VARCHAR(5), FOREIGN KEY (FAC_ID) REFERENCES FACULTY, FOREIGN KEY (SUB_ID) REFERENCES SUBJECT);



CREATE TABLE STUDENT(STUD_ID VARCHAR(5) PRIMARY KEY, STUD_NAME VARCHAR(20) NOT NULL, GENDER VARCHAR(10), CGPA INTEGER, AGE INTEGER CHECK (AGE>=18));



CREATE TABLE STUDLOG(STUD_ID VARCHAR(5) PRIMARY KEY, STUD_NAME VARCHAR(20) NOT NULL, GENDER VARCHAR(10), CGPA INTEGER, AGE INTEGER CHECK (AGE>=18));

SQL Worksheet

```sql
CREATE TABLE DEPARTMENT(DEP_ID VARCHAR(5) PRIMARY KEY, DEP_NAME VARCHAR(20)  NOT NULL);

CREATE TABLE FACULTY(FAC_ID VARCHAR(5) PRIMARY KEY, FAC_NAME VARCHAR(20) NOT NULL, ROOM_NO INTEGER, DEP_ID VARCHAR(5), FOREIGN KEY (DEP_ID) REFERENCES DEPARTMENT);

CREATE TABLE SUBJECT(SUB_ID VARCHAR(5) PRIMARY KEY, SUB_NAME VARCHAR(20), FAC_ID VARCHAR(5), FOREIGN KEY (FAC_ID) REFERENCES FACULTY,
DEP_ID VARCHAR(5), FOREIGN KEY (DEP_ID) REFERENCES DEPARTMENT);

CREATE TABLE TEACHES(FAC_ID VARCHAR(5), SUB_ID VARCHAR(5), FOREIGN KEY (FAC_ID) REFERENCES FACULTY, FOREIGN KEY (SUB_ID) REFERENCES SUBJECT);

CREATE TABLE STUDENT(STUD_ID VARCHAR(5) PRIMARY KEY, STUD_NAME VARCHAR(20) NOT NULL, GENDER VARCHAR(10), CGPA INTEGER, AGE INTEGER CHECK (AGE>=18));

CREATE TABLE STUDLOG(STUD_ID VARCHAR(5) PRIMARY KEY, STUD_NAME VARCHAR(20) NOT NULL, GENDER VARCHAR(10), CGPA INTEGER, AGE INTEGER CHECK (AGE>=18));
```

Table created.

CREATE TABLE ENROLLED(STUD_ID VARCHAR(5), DEP_ID VARCHAR(5), PRIMARY KEY(STUD_ID, DEP_ID), FOREIGN KEY (DEP_ID) REFERENCES DEPARTMENT,
FOREIGN KEY (STUD_ID) REFERENCES STUDENT);

SQL Worksheet

```sql
CREATE TABLE ENROLLED(STUD_ID VARCHAR(5), DEP_ID VARCHAR(5), PRIMARY KEY(STUD_ID, DEP_ID), FOREIGN KEY (DEP_ID) REFERENCES DEPARTMENT,
FOREIGN KEY (STUD_ID) REFERENCES STUDENT);

CREATE TABLE EXAM(EXAM_ID VARCHAR(5) PRIMARY KEY, SUB_ID VARCHAR(5), FOREIGN KEY (SUB_ID) REFERENCES SUBJECT);

CREATE TABLE GRADES(STUD_ID VARCHAR(5), EXAM_ID VARCHAR(5), PRIMARY KEY(STUD_ID, EXAM_ID), FOREIGN KEY (EXAM_ID) REFERENCES EXAM,
FOREIGN KEY (STUD_ID) REFERENCES STUDENT, MARKS INTEGER, GRADE VARCHAR(2));

CREATE TABLE ACCOUNTS(STUD_ID VARCHAR(5), FEE INTEGER, STATUS VARCHAR(3), FOREIGN KEY (STUD_ID) REFERENCES STUDENT);

INSERT INTO UNIVERSITY VALUES('B01', 'GITAM, VSKP', 'VISAKHAPATNAM');
INSERT INTO UNIVERSITY VALUES('B02', 'GITAM, BANGLORE', 'BANGLORE');
INSERT INTO UNIVERSITY VALUES('B03', 'GITAM, HYD', 'HYDERABAD');
SELECT * FROM UNIVERSITY;
```

Table created.

CREATE TABLE EXAM(EXAM_ID VARCHAR(5) PRIMARY KEY, SUB_ID VARCHAR(5), FOREIGN KEY (SUB_ID) REFERENCES SUBJECT);

SQL Worksheet

Clear Find Actions Save Run

```
CREATE TABLE ENROLLED(STUD_ID VARCHAR(5), DEP_ID VARCHAR(5), PRIMARY KEY(STUD_ID, DEP_ID), FOREIGN KEY (DEP_ID) REFERENCES DEPARTMENT,
FOREIGN KEY (STUD_ID) REFERENCES STUDENT);

CREATE TABLE EXAM(EXAM_ID VARCHAR(5) PRIMARY KEY, SUB_ID VARCHAR(5), FOREIGN KEY (SUB_ID) REFERENCES SUBJECT);

CREATE TABLE GRADES(STUD_ID VARCHAR(5), EXAM_ID VARCHAR(5), PRIMARY KEY(STUD_ID, EXAM_ID), FOREIGN KEY (EXAM_ID) REFERENCES EXAM,
FOREIGN KEY (STUD_ID) REFERENCES STUDENT, MARKS INTEGER, GRADE VARCHAR(2));

CREATE TABLE ACCOUNTS(STUD_ID VARCHAR(5), FEE INTEGER, STATUS VARCHAR(3), FOREIGN KEY (STUD_ID) REFERENCES STUDENT);

INSERT INTO UNIVERSITY VALUES('B01', 'GITAM, VSKP', 'VISAKHAPATNAM');
INSERT INTO UNIVERSITY VALUES('B02', 'GITAM, BANGLORE', 'BANGLORE');
INSERT INTO UNIVERSITY VALUES('B03', 'GITAM, HYD', 'HYDERABAD');
SELECT * FROM UNIVERSITY;
```

Table created.

CREATE TABLE GRADES(STUD_ID VARCHAR(5), EXAM_ID VARCHAR(5), PRIMARY KEY(STUD_ID, EXAM_ID), FOREIGN KEY (EXAM_ID) REFERENCES EXAM,
FOREIGN KEY (STUD_ID) REFERENCES STUDENT, MARKS INTEGER, GRADE VARCHAR(2));

SQL Worksheet

Clear Find Actions Save Run

```
CREATE TABLE ENROLLED(STUD_ID VARCHAR(5), DEP_ID VARCHAR(5), PRIMARY KEY(STUD_ID, DEP_ID), FOREIGN KEY (DEP_ID) REFERENCES DEPARTMENT,
FOREIGN KEY (STUD_ID) REFERENCES STUDENT);

CREATE TABLE EXAM(EXAM_ID VARCHAR(5) PRIMARY KEY, SUB_ID VARCHAR(5), FOREIGN KEY (SUB_ID) REFERENCES SUBJECT);

CREATE TABLE GRADES(STUD_ID VARCHAR(5), EXAM_ID VARCHAR(5), PRIMARY KEY(STUD_ID, EXAM_ID), FOREIGN KEY (EXAM_ID) REFERENCES EXAM,
FOREIGN KEY (STUD_ID) REFERENCES STUDENT, MARKS INTEGER, GRADE VARCHAR(2));

CREATE TABLE ACCOUNTS(STUD_ID VARCHAR(5), FEE INTEGER, STATUS VARCHAR(3), FOREIGN KEY (STUD_ID) REFERENCES STUDENT);

INSERT INTO UNIVERSITY VALUES('B01', 'GITAM, VSKP', 'VISAKHAPATNAM');
INSERT INTO UNIVERSITY VALUES('B02', 'GITAM, BANGLORE', 'BANGLORE');
INSERT INTO UNIVERSITY VALUES('B03', 'GITAM, HYD', 'HYDERABAD');
SELECT * FROM UNIVERSITY;
```

Table created.

CREATE TABLE ACCOUNTS(STUD_ID VARCHAR(5), FEE INTEGER, STATUS VARCHAR(3), FOREIGN KEY (STUD_ID) REFERENCES STUDENT);

SQL Worksheet

```
CREATE TABLE GRADES(STUD_ID VARCHAR(5), EXAM_ID VARCHAR(5), PRIMARY KEY(STUD_ID, EXAM_ID), FOREIGN KEY (EXAM_ID) REFERENCES EXAM,
FOREIGN KEY (STUD_ID) REFERENCES STUDENT, MARKS INTEGER, GRADE VARCHAR(2));

CREATE TABLE ACCOUNTS(STUD_ID VARCHAR(5), FEE INTEGER, STATUS VARCHAR(3), FOREIGN KEY (STUD_ID) REFERENCES STUDENT);

INSERT INTO UNIVERSITY VALUES('B01', 'GITAM, VSKP', 'VISAKHAPATNAM');
INSERT INTO UNIVERSITY VALUES('B02', 'GITAM, BANGLORE', 'BANGLORE');
INSERT INTO UNIVERSITY VALUES('B03', 'GITAM, HYD', 'HYDERABAD');
SELECT * FROM UNIVERSITY;

INSERT INTO DEPARTMENT VALUES('D01', 'CSE');
INSERT INTO DEPARTMENT VALUES('D02', 'MECH');
INSERT INTO DEPARTMENT VALUES('D03', 'DENTAL');
INSERT INTO DEPARTMENT VALUES('D04', 'LAW');
```

Table created.

Collect data manually and create Database /Insert the values

INSERT INTO UNIVERSITY VALUES('B01', 'GITAM, VSKP', 'VISAKHAPATNAM');
INSERT INTO UNIVERSITY VALUES('B02', 'GITAM, BANGLORE', 'BANGLORE');
INSERT INTO UNIVERSITY VALUES('B03', 'GITAM, HYD', 'HYDERABAD');
SELECT * FROM UNIVERSITY;

SQL Worksheet

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

BRANCH_ID	BRANCH_NAME	BRANCH_ADDRESS
B01	GITAM, VSKP	VISAKHAPATNAM
B02	GITAM, BANGLORE	BANGLORE
B03	GITAM, HYD	HYDERABAD

Download CSV
3 rows selected.

INSERT INTO DEPARTMENT VALUES('D01', 'CSE'); INSERT INTO DEPARTMENT VALUES('D02', 'MECH');
INSERT INTO DEPARTMENT VALUES('D03', 'DENTAL');
INSERT INTO DEPARTMENT VALUES('D04', 'LAW'); INSERT INTO DEPARTMENT VALUES('D05', 'ECE'); INSERT INTO DEPARTMENT VALUES('D06', 'PHARMACY');
INSERT INTO DEPARTMENT VALUES('D07', 'EEE'); INSERT INTO DEPARTMENT VALUES('D08', 'MEDICAL');
INSERT INTO DEPARTMENT VALUES('D09', 'ARCH'); SELECT * FROM DEPARTMENT;

DEP_ID	DEP_NAME
D01	CSE
D02	MECH
D03	DENTAL
D04	LAW
D05	ECE
D06	PHARMACY
D07	EEE
D08	MEDICAL
D09	ARCH

INSERT INTO FACULTY VALUES('F01', 'PREM SINGH', '1', 'D01');
INSERT INTO FACULTY VALUES('F02', 'MANSA DEVI', '1', 'D01');
INSERT INTO FACULTY VALUES('F03', 'BHAVANI', '2', 'D02');
INSERT INTO FACULTY VALUES('F04', 'SRINIVAS RAO', '2', 'D02');
INSERT INTO FACULTY VALUES('F05', 'VIJAY SINGH', '3', 'D03');
INSERT INTO FACULTY VALUES('F06', 'PRAVEEN KUMAR', '3', 'D03');
INSERT INTO FACULTY VALUES('F07', 'VANDANA SINGH', '4', 'D04');
INSERT INTO FACULTY VALUES('F08', 'RAGHAVENDRA', '4', 'D04');
INSERT INTO FACULTY VALUES('F09', 'GHANSHYAM', '5', 'D05');
INSERT INTO FACULTY VALUES('F10', 'ABHISHEK GV', '5', 'D05');
INSERT INTO FACULTY VALUES('F11', 'SASI KUMAR', '6', 'D06');
INSERT INTO FACULTY VALUES('F12', 'L GONDI', '6', 'D06');
INSERT INTO FACULTY VALUES('F13', 'DON S', '7', 'D07');
INSERT INTO FACULTY VALUES('F14', 'RAVI', '7', 'D07');
INSERT INTO FACULTY VALUES('F15', 'GRACE', '8', 'D08');

```
INSERT INTO FACULTY VALUES('F16', 'K KAVITA', '8', 'D08');
INSERT INTO FACULTY VALUES('F17', 'ANKIREDDY', '9',
'D09');
INSERT INTO FACULTY VALUES('F18', 'MURALI MOHAN', '9', 'D09');
INSERT INTO FACULTY VALUES('F19', 'SUKHIBHAVA', '2', 'D02');
INSERT INTO FACULTY VALUES('F20', 'TANGUTOORI', '3',
'D03');
INSERT INTO FACULTY VALUES('F21', 'JAYA SRI', '8', 'D08');
INSERT INTO FACULTY VALUES('F22', 'HIMAJA', '4', 'D04');
INSERT INTO FACULTY VALUES('F23', 'LUCKY', '9', 'D09');
SELECT * FROM FACULTY;
```

SQL Worksheet

Clear Find Actions Save Run

FAC_ID	FAC_NAME	ROOM_NO	DEP_ID
F01	PREM SINGH	1	D01
F02	MANSA DEVI	1	D01
F03	BHAVANI	2	D02
F04	SRINIVAS RAO	2	D02
F05	VIJAY SINGH	3	D03
F06	PRAVEEN KUMAR	3	D03
F07	VANDANA SINGH	4	D04
F08	RAGHAVENDRA	4	D04
F09	GHANSHYAM	5	D05
F10	ABHISHEK GV	5	D05
F11	SASI KUMAR	6	D06
F12	L GONDI	6	D06
F13	DON S	7	D07

```
INSERT INTO SUBJECT VALUES('S01', 'ADV DATA STRUCTURES', 'F01',
'D01');
INSERT INTO SUBJECT VALUES('S02', 'DBMS', 'F02', 'D01');
INSERT INTO SUBJECT VALUES('S03', 'THERMO DYNAMICS', 'F03', 'D02');
INSERT INTO SUBJECT VALUES('S04', 'PHYSICS', 'F04', 'D02');
INSERT INTO SUBJECT VALUES('S05', 'TEETH', 'F05', 'D03');
```

```
INSERT INTO SUBJECT VALUES('S06', 'GUMS', 'F06', 'D03');
INSERT INTO SUBJECT VALUES('S07', 'CIVICS', 'F07', 'D04');
INSERT INTO SUBJECT VALUES('S08', 'CRIMINOLOGY',
'F08', 'D04');
INSERT INTO SUBJECT VALUES('S09', 'FDLC', 'F09', 'D05');
INSERT INTO SUBJECT VALUES('S10', 'COA', 'F10', 'D05');
INSERT INTO SUBJECT VALUES('S11', 'LIFE SCIENCES', 'F11',
'D06');
INSERT INTO SUBJECT VALUES('S12', 'BIO CHEMISTRY', 'F11',
'D06');
INSERT INTO SUBJECT VALUES('S13', 'BEEE', 'F13', 'D07');
INSERT INTO SUBJECT VALUES('S14', 'ELECTRICAL',
'F14', 'D07');
INSERT INTO SUBJECT VALUES('S15', 'HUMAN BODY', 'F15', 'D08');
INSERT INTO SUBJECT VALUES('S16', 'REPRODUCTION',
'F16', 'D08');
INSERT INTO SUBJECT VALUES('S17', 'SCALES', 'F17', 'D09');
INSERT INTO SUBJECT VALUES('S18', 'AUTOCAD', 'F18', 'D09');
SELECT * FROM SUBJECT;
```

SQL Worksheet

Clear Find Actions Save Run

SUB_ID	SUB_NAME	FAC_ID	DEP_ID
S01	ADV DATA STRUCTURES	F01	D01
S02	DBMS	F02	D01
S03	THERMO DYNAMICS	F03	D02
S04	PHYSICS	F04	D02
S05	TEETH	F05	D03
S06	GUMS	F06	D03
S07	CIVICS	F07	D04
S08	CRIMINOLOGY	F08	D04
S09	FDLC	F09	D05
S10	COA	F10	D05
S11	LIFE SCIENCES	F11	D06
S12	BIO CHEMISTRY	F11	D06
S13	BEEE	F13	D07

```
INSERT INTO STUDENT VALUES('ST01', 'ABHISHEK',
'MALE', 9, 20);
INSERT INTO STUDENT VALUES('ST02', 'VISHNU', 'MALE', 8, 20);
INSERT INTO STUDENT VALUES('ST03', 'KAMALA', 'FEMALE', 8, 19);
INSERT INTO STUDENT VALUES('ST04', 'JAYA', 'FEMALE', 1, 18);
INSERT INTO STUDENT VALUES('ST05', 'SAKETH', 'MALE', 9, 18);
INSERT INTO STUDENT VALUES('ST06', 'SOHAN', 'MALE', 7, 19);
INSERT INTO STUDENT VALUES('ST07', 'PRANAY', 'MALE', 6, 19);
INSERT INTO STUDENT VALUES('ST08', 'AKHILESH',
'MALE', 7, 20);
INSERT INTO STUDENT VALUES('ST09', 'ANVESH', 'MALE', 6, 20);
INSERT INTO STUDENT VALUES('ST10', 'PRATHEK', 'MALE', 10, 21);
SELECT * FROM STUDENT;
```

SQL Worksheet

Clear Find Actions Save Run

STUD_ID	STUD_NAME	GENDER	CGPA	AGE
ST01	ABHISHEK	MALE	9	20
ST02	VISHNU	MALE	8	20
ST03	KAMALA	FEMALE	8	19
ST04	JAYA	FEMALE	1	18
ST05	SAKETH	MALE	9	18
ST06	SOHAN	MALE	7	19
ST07	PRANAY	MALE	6	19
ST08	AKHILESH	MALE	7	20
ST09	ANVESH	MALE	6	20
ST10	PRATHEK	MALE	10	21

Download CSV

10 rows selected.

```
INSERT INTO TEACHES VALUES('F01', 'S01');
INSERT INTO TEACHES VALUES('F02', 'S02');
 INSERT INTO TEACHES VALUES('F03', 'S03');
 INSERT INTO TEACHES VALUES('F04', 'S04');
INSERT INTO TEACHES VALUES('F05', 'S05');
 INSERT INTO TEACHES VALUES('F06', 'S06');
 INSERT INTO TEACHES VALUES('F07', 'S07');
 INSERT INTO TEACHES VALUES('F08', 'S08');
 INSERT INTO TEACHES VALUES('F09', 'S09');
 INSERT INTO TEACHES VALUES('F10', 'S10');
 INSERT INTO TEACHES VALUES('F11', 'S11');
 INSERT INTO TEACHES VALUES('F12', 'S12');
 INSERT INTO TEACHES VALUES('F13', 'S13');
 INSERT INTO TEACHES VALUES('F14', 'S14');
 INSERT INTO TEACHES VALUES('F15', 'S15');
 INSERT INTO TEACHES VALUES('F16', 'S16');
 INSERT INTO TEACHES VALUES('F17', 'S17');
INSERT INTO TEACHES VALUES('F18', 'S18');
SELECT* FROM TEACHES;
```

FAC_ID	SUB_ID
F01	S01
F02	S02
F03	S03
F04	S04
F05	S05
F06	S06
F07	S07
F08	S08
F09	S09
F10	S10
F11	S11
F12	S12
F13	S13

```
INSERT INTO ENROLLED VALUES('ST01', 'D01');
 INSERT INTO ENROLLED VALUES('ST02', 'D02');
 INSERT INTO ENROLLED VALUES('ST03', 'D03');
INSERT INTO ENROLLED VALUES('ST04', 'D04');
 INSERT INTO ENROLLED VALUES('ST05', 'D05');
INSERT INTO ENROLLED VALUES('ST06', 'D06');
INSERT INTO ENROLLED VALUES('ST07', 'D07');
INSERT INTO ENROLLED VALUES('ST08', 'D08');
```

```
INSERT INTO ENROLLED VALUES('ST09', 'D09');
INSERT INTO ENROLLED VALUES('ST10', 'D01');
INSERT INTO ENROLLED VALUES('ST04', 'D01');
INSERT INTO ENROLLED VALUES('ST05', 'D07');
INSERT INTO ENROLLED VALUES('ST06', 'D08');
SELECT* FROM ENROLLED;
```

STUD_ID	DEP_ID
ST01	D01
ST02	D02
ST03	D03
ST04	D01
ST04	D04
ST05	D05
ST05	D07
ST06	D06
ST06	D08
ST07	D07
ST08	D08
ST09	D09

```
INSERT INTO EXAM VALUES('E01', 'S01');
 INSERT INTO EXAM VALUES('E02', 'S02');
INSERT INTO EXAM VALUES('E03', 'S03');
INSERT INTO EXAM VALUES('E04', 'S04');
INSERT INTO EXAM VALUES('E05', 'S05');
INSERT INTO EXAM VALUES('E06', 'S06');
 INSERT INTO EXAM VALUES('E07', 'S07');
 INSERT INTO EXAM VALUES('E08', 'S08');
INSERT INTO EXAM VALUES('E09', 'S09');
INSERT INTO EXAM VALUES('E10', 'S10');
 INSERT INTO EXAM VALUES('E11', 'S11');
 INSERT INTO EXAM VALUES('E12', 'S12');
INSERT INTO EXAM VALUES('E13', 'S13');
INSERT INTO EXAM VALUES('E14', 'S14');
 INSERT INTO EXAM VALUES('E15', 'S15');
INSERT INTO EXAM VALUES('E16', 'S16');
INSERT INTO EXAM VALUES('E17', 'S17');
INSERT INTO EXAM VALUES('E18', 'S18');
SELECT * FROM EXAM;
```

EXAM_ID	SUB_ID
E01	S01
E02	S02
E03	S03
E04	S04
E05	S05
E06	S06
E07	S07
E08	S08
E09	S09
E10	S10
E11	S11
E12	S12
E13	S13

```
INSERT INTO GRADES VALUES('ST01', 'E01' ,95, 'A');
INSERT INTO GRADES VALUES('ST01', 'E02' ,85, 'B');
INSERT INTO GRADES VALUES('ST02', 'E03' ,95, 'A');
INSERT INTO GRADES VALUES('ST02', 'E04' ,85, 'B');
 INSERT INTO GRADES VALUES('ST03', 'E05' ,75, 'C');
 INSERT INTO GRADES VALUES('ST03', 'E06' ,75, 'C');
 INSERT INTO GRADES VALUES('ST04', 'E07' ,75, 'C');
INSERT INTO GRADES VALUES('ST04', 'E08' ,95, 'A');
 INSERT INTO GRADES VALUES('ST05', 'E09' ,85, 'B');
INSERT INTO GRADES VALUES('ST05', 'E10' ,85, 'B');
INSERT INTO GRADES VALUES('ST06', 'E11' ,95, 'A');
INSERT INTO GRADES VALUES('ST06', 'E12' ,65, 'D');
 INSERT INTO GRADES VALUES('ST07', 'E13' ,65, 'D');
INSERT INTO GRADES VALUES('ST07', 'E14' ,95, 'A');
 INSERT INTO GRADES VALUES('ST08', 'E15' ,75, 'C');
 INSERT INTO GRADES VALUES('ST08', 'E16' ,85, 'B');
INSERT INTO GRADES VALUES('ST09', 'E17' ,85, 'B');
INSERT INTO GRADES VALUES('ST10', 'E18' ,95, 'A');
SELECT * FROM GRADES;
```

SQL Worksheet

Clear Find Actions Save Run

STUD_ID	EXAM_ID	MARKS	GRADE
ST01	E01	95	A
ST01	E02	85	B
ST02	E03	95	A
ST02	E04	85	B
ST03	E05	75	C
ST03	E06	75	C
ST04	E07	75	C
ST04	E08	95	A
ST05	E09	85	B
ST05	E10	85	B
ST06	E11	95	A
ST06	E12	65	D
ST07	E13	65	D
ST07	E14	95	A

```
INSERT INTO ACCOUNTS VALUES('ST01', 270000, 'YES');
INSERT INTO ACCOUNTS VALUES('ST02', 250000, 'YES');
INSERT INTO ACCOUNTS VALUES('ST03', 250000, 'NO');
INSERT INTO ACCOUNTS VALUES('ST04', 210000, 'NO');
INSERT INTO ACCOUNTS VALUES('ST05', 170000, 'YES');
INSERT INTO ACCOUNTS VALUES('ST06', 350000, 'YES');
INSERT INTO ACCOUNTS VALUES('ST07', 320000, 'NO');
INSERT INTO ACCOUNTS VALUES('ST08', 270000, 'YES');
INSERT INTO ACCOUNTS VALUES('ST09', 270000, 'YES');
INSERT INTO ACCOUNTS VALUES('ST10', 220000, 'NO');
SELECT * FROM ACCOUNTS;
```

SQL Worksheet

Clear Find Actions Save Run

1 row(s) inserted.

STUD_ID	FEE	STATUS
ST01	270000	YES
ST02	250000	YES
ST03	250000	NO
ST04	210000	NO
ST05	170000	YES
ST06	350000	YES
ST07	320000	NO
ST08	270000	YES
ST09	270000	YES
ST10	220000	NO

Download CSV

10 rows selected.

- **Perform all the Various SQL Commands Operations.**

--ALTER MODIFY

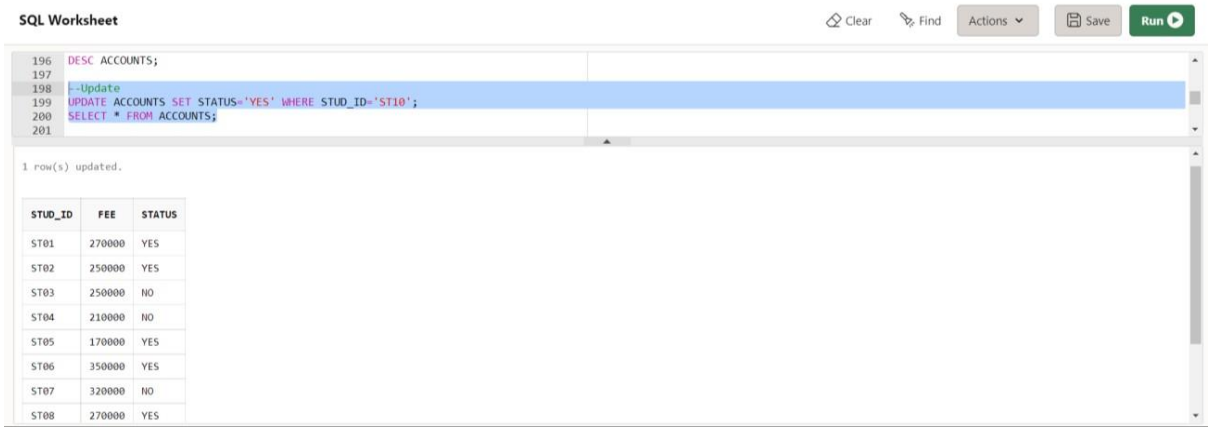
ALTER TABLE ACCOUNTS MODIFY STATUS VARCHAR(3) NOT NULL;

DESC ACCOUNTS;

-UPDATE

UPDATE ACCOUNTS SET STATUS='YES' WHERE STUD_ID='ST10';

SELECT * FROM ACCOUNTS;

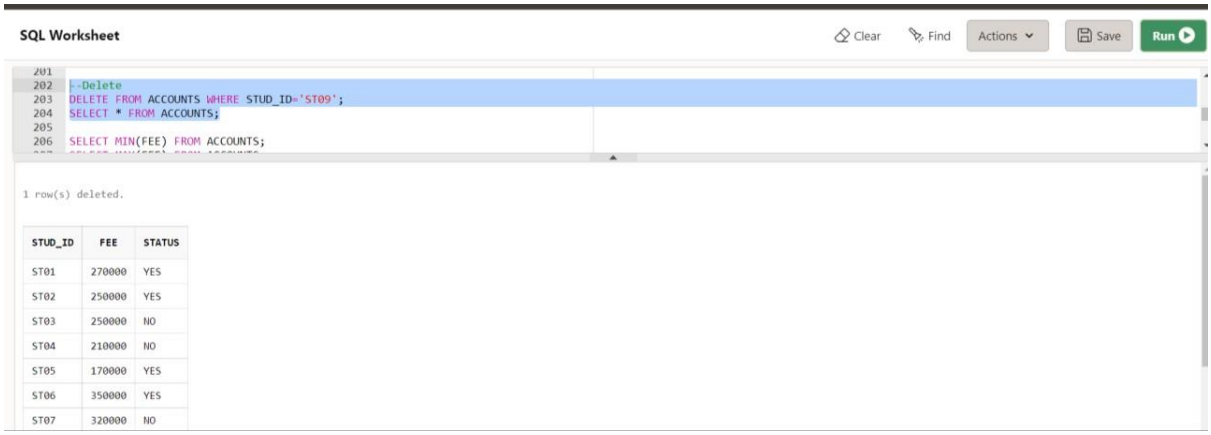


--DELETE

DELETE FROM ACCOUNTS WHERE

STUD_ID='ST09';

SELECT * FROM ACCOUNTS;



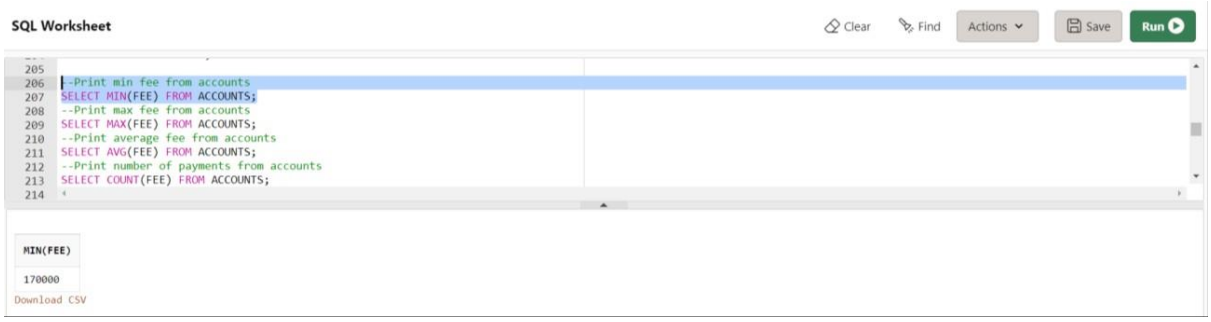
Part-IV Queries:

Write Queries Related To Database

1. Aggregate Operators

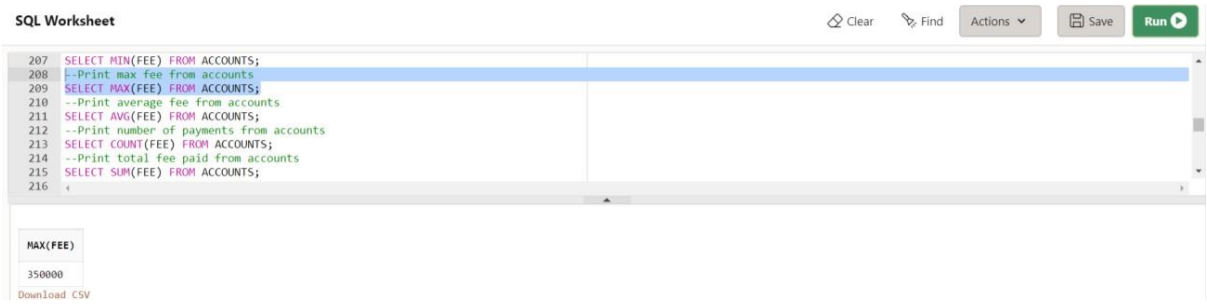
--PRINT MIN FEE FROM ACCOUNTS

SELECT MIN(FEE) FROM ACCOUNTS;



--PRINT MAX FEE FROM ACCOUNTS

SELECT MAX(FEE) FROM ACCOUNTS;



--PRINT AVERAGE FEE FROM ACCOUNTS

SELECT AVG(FEE) FROM ACCOUNTS;

--PRINT NUMBER OF PAYMENTS FROM ACCOUNTS

SELECT COUNT(FEE) FROM ACCOUNTS;

--PRINT TOTAL FEE PAID FROM ACCOUNTS

SELECT SUM(FEE) FROM ACCOUNTS;

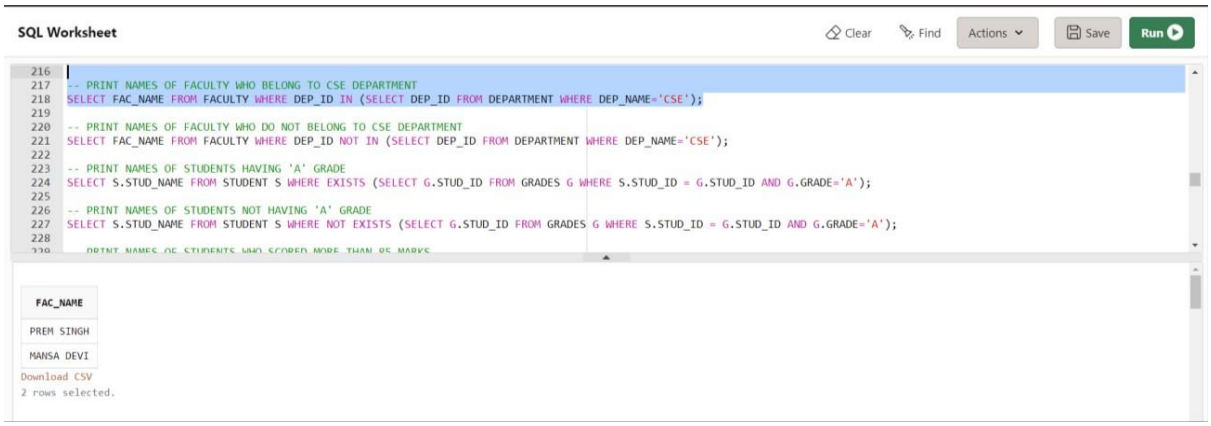
2. Nested Queries

-- PRINT NAMES OF FACULTY WHO BELONG TO CSE DEPARTMENT

SELECT FAC_NAME FROM FACULTY WHERE

DEP_ID IN (SELECT DEP_ID FROM DEPARTMENT

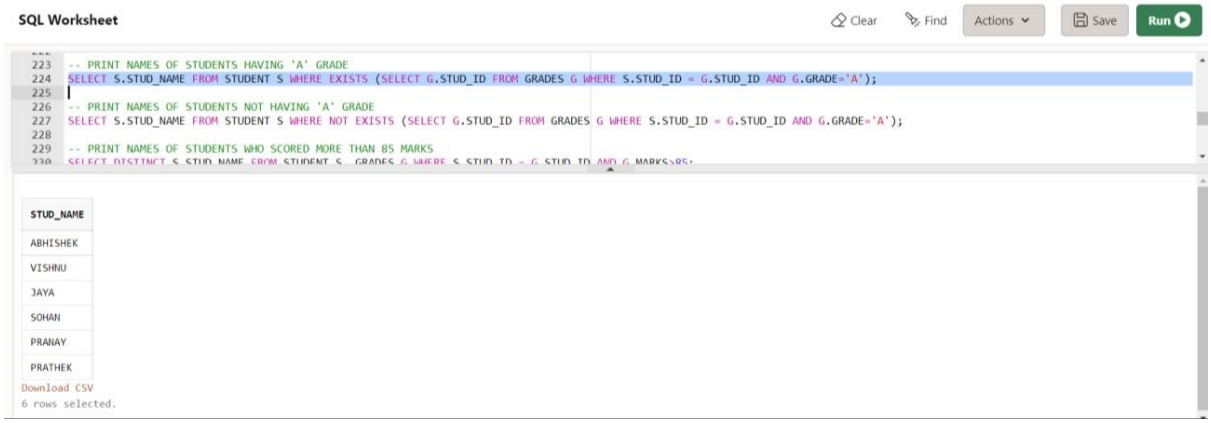
WHERE DEP_NAME='CSE');



3. Correlated Nested Query

-- PRINT NAMES OF STUDENTS HAVING 'A' GRADE

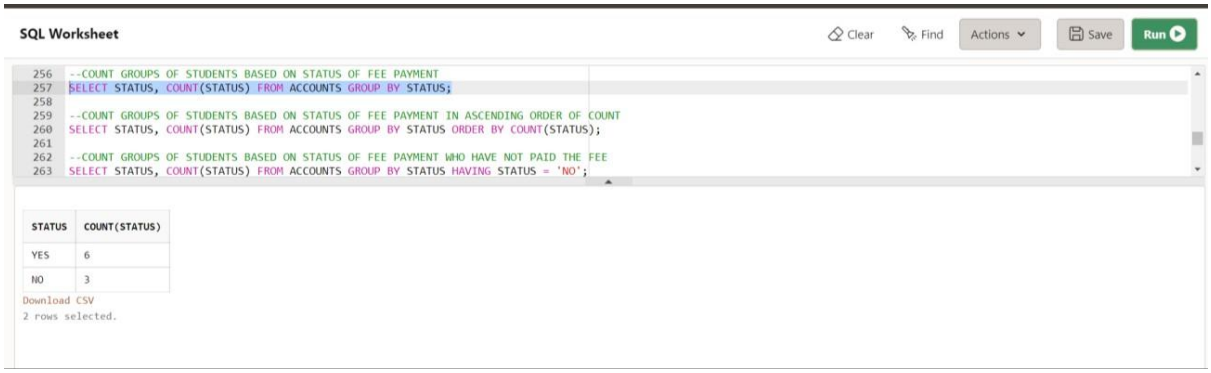
SELECT S.STUD_NAME FROM STUDENT S WHERE EXISTS(SELECTG.STUD_ID FROM GRADES G WHERE S.STUD_ID = G.STUD_ID AND G.GRADE='A');



4. Group by Having

--COUNT GROUPS OF STUDENTS BASED ON STATUS OF FEE PAYMENT

SELECT STATUS, COUNT(STATUS) FROM ACCOUNTS GROUP BY STATUS;



5. Relational SET Queries

--PRINT NAMES OF STUDENTS WHO ENROLLED IN EITHER 'CSE' OR 'EEE'

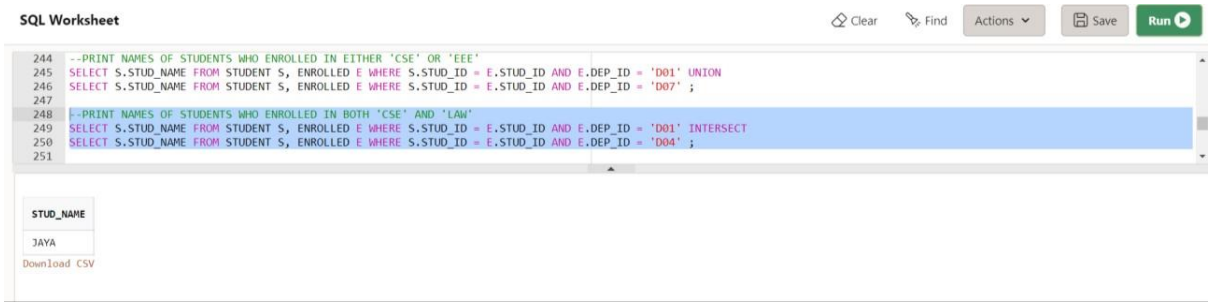
SELECT S.STUD_NAME FROM STUDENT S, ENROLLED E WHERE S.STUD_ID = E.STUD_ID AND E.DEP_ID = 'D01' UNION

SELECT S.STUD_NAME FROM STUDENT S, ENROLLED E WHERE S.STUD_ID = E.STUD_ID AND E.DEP_ID = 'D07' ;

--PRINT NAMES OF STUDENTS WHO ENROLLED IN BOTH 'CSE' AND 'LAW'

SELECT S.STUD_NAME FROM STUDENT S, ENROLLED E WHERE S.STUD_ID = E.STUD_ID AND E.DEP_ID = 'D01' INTERSECT

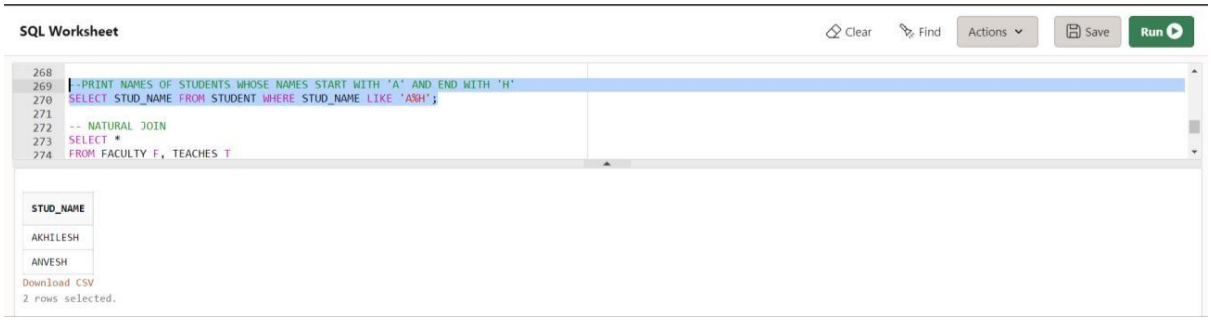
SELECT S.STUD_NAME FROM STUDENT S, ENROLLED E WHERE S.STUD_ID = E.STUD_ID AND E.DEP_ID = 'D04' ;



6. Between, Like operator

--PRINT NAMES OF STUDENTS WHOSE NAMES START WITH 'A' AND END WITH 'H'

SELECT STUD_NAME FROM STUDENT WHERE STUD_NAME LIKE 'A%H';



7. Create Triggers For The Database And Cursors

-- BEFORE INSERT

-- CREATE TRIGGER TO CHECK THAT AGE MUST BE ATLEAST 18 YEARS BEFORE INSERTING

CREATE TRIGGER CHECKAGE18

BEFORE INSERT ON STUDENT

FOR EACH ROW

WHEN (NEW.AGE<18)

BEGIN

RAISE_APPLICATION_ERROR(-2000, 'ERROR! AGE MUST BE ATLEAST 18 YEARS!');

END;

INSERT INTO STUDENT VALUES('ST01', 'ABHISHEK', 'MALE', 9, 15);

-- BEFORE DELETE

-- CREATE TRIGGER TO CHECK THAT AGE MUST NOT BE GREATER THAN 20 YEARS BEFORE DELETING
CREATE TRIGGER CHECKAGE20

BEFORE DELETE ON STUDENT

FOR EACH ROW

WHEN (OLD.AGE>20)

BEGIN

RAISE_APPLICATION_ERROR(-2000, 'ERROR! AGE MUST NOT BE GREATER THAN 20 YEARS!');

END;

DELETE FROM STUDENT WHERE STUD_ID = 'ST10';

-- AFTER UPDATE

--CREATE TRIGGER TO STORE STUDENT UPDATE LOG AFTER UPDATING IF AGE IS GREATER THAN OR EQUAL TO 18

CREATE TRIGGER CHECKAGE

AFTER UPDATE ON STUDENT

FOR EACH ROW

WHEN (OLD.AGE>18)

BEGIN

INSERT INTO STUDLOG VALUES(:NEW.STUD_ID,

:NEW.STUD_NAME, :NEW.GENDER, :NEW.CGPA,

:NEW.AGE

END;

UPDATE STUDENT SET AGE=23 WHERE

STUD_ID='ST04';

SELECT * FROM STUD;

-- CURSOR

-- RETRIEVE DATA FROM STUDENT TABLE AND PRINT IN TEXT FORMAT

DECLARE

CURSOR Q10 IS SELECT * FROM STUDENT;

Q20 Q10 % ROWTYPE;

BEGIN

OPEN Q10;

DBMS_OUTPUT.PUT_LINE('DETAILS OF STUDENTS');

LOOP

FETCH Q10 INTO Q20;

EXIT WHEN (Q10 % NOTFOUND);

DBMS_OUTPUT.PUT_LINE('STUDENT ID: '||Q20.STUD_ID||' STUDENT NAME: '||Q20.STUD_NAME||' AGE: '||Q20.AGE||' CGPA: '||Q20.CGPA||' GENDER: '||Q20.GENDER);

END LOOP;

CLOSE Q10;

END;

SQL Worksheet

```
-- CURSOR
DECLARE
CURSOR Q10 IS SELECT * FROM STUDENT;
Q20 Q10 % ROWTYPE;
BEGIN
OPEN Q10;
DBMS_OUTPUT.PUT_LINE( 'DETAILS OF STUDENTS' );
LOOP
FETCH Q10 INTO Q20;
EXIT WHEN (Q10 % NOTFOUND);
DBMS_OUTPUT.PUT_LINE('STUDENT ID: '||Q20.STUD_ID||' STUDENT NAME: '||Q20.STUD_NAME||' AGE: '||Q20.AGE||' CGPA: '||Q20.CGPA||' GENDER: '||Q20.GENDER);
END LOOP;
CLOSE Q10;
END;
```

```
Statement processed.
DETAILS OF STUDENTS
STUDENT ID: ST01 STUDENT NAME: ABHISHEK AGE: 20 CGPA: 9 GENDER: MALE
STUDENT ID: ST02 STUDENT NAME: VISHNU AGE: 20 CGPA: 8 GENDER: MALE
STUDENT ID: ST03 STUDENT NAME: KAMALA AGE: 23 CGPA: 8 GENDER: FEMALE
STUDENT ID: ST04 STUDENT NAME: JAYA AGE: 23 CGPA: 1 GENDER: FEMALE
STUDENT ID: ST05 STUDENT NAME: SAKETH AGE: 18 CGPA: 9 GENDER: MALE
STUDENT ID: ST06 STUDENT NAME: SOHAN AGE: 19 CGPA: 7 GENDER: MALE
STUDENT ID: ST07 STUDENT NAME: PRANAY AGE: 19 CGPA: 6 GENDER: MALE
STUDENT ID: ST08 STUDENT NAME: AKHILESH AGE: 20 CGPA: 7 GENDER: MALE
STUDENT ID: ST09 STUDENT NAME: ANVESH AGE: 20 CGPA: 6 GENDER: MALE
```

PART – 5 NORMALIZATION –

For normalization we will consider the tables FACULTY

FAC_ID	FAC_NAME	ROOM_NO	DEP_ID
F01	PREM SINGH	1	D01
F02	MANSA DEVI	1	D01
F03	BHAVANI	2	D02
F04	SRINIVAS RAO	2	D02
F05	VIJAY SINGH	3	D03
F06	PRAVEEN KUMAR	3	D03
F07	VANDANA SINGH	4	D04
F08	RAGHAVENDRA	4	D04
F09	GHANSHYAM	5	D05
F10	ABHISHEK GV	5	D05
F11	SASI KUMAR	6	D06
F12	L GONDI	6	D06
F13	DON S	7	D07
F14	RAVI	7	D07
F15	GRACE	8	D08
F16	K KAVITA	8	D08
F17	ANKIREDDY	9	D09
F18	MURALI MOHAN	9	D09
F19	SUKHIBHAVA	2	D02
F20	TANGUTOORI	3	D03
F21	JAYA SRI	8	D08
F22	HIMAJA	4	D04
F23	LUCKY	9	D09

Download CSV

FACULTY(FAC_ID(PK), FAC_NAME, ROOM_NO, DEP_ID)

FUNCTIONAL DEPENDACIES:

{

FAC_ID → FAC_ID, FAC_NAME

DEP_ID → DEP_ID, ROOM_NO

}

CANDIDATE KEY:

THE CANDIDATE KEY WILL BE: (FAC_ID, DEP_ID)

BECAUSE BY PERFORMING ATTRIBUTE CLOSURE: (FAC_ID)+ = (FAC_ID, FAC_NAME)

(DEP_ID)+ = (DEP_ID, ROOM_NO)

(FAC_ID, DEP_ID)+ = (FAC_ID, FAC_NAME, DEP_ID, ROOM_NO)

ALL ATTRIBUTES ARE DERIVED THEREFORE CK = (FAC_ID, DEP_ID)

PRIME ATTRIBUTES ARE THE ATTRIBUTES PRESENT IN CK-

- PRIME ATTRIBUTES: (FAC_ID, DEP_ID)
- NON-PRIME ATTRIBUTES: (FAC_NAME, ROOM_NO)

<u>1st Normal form:</u>

The faculty relation is already in 1NF

<u>2nd Normal form:</u>

LET US CHECK FOR FACULTY RELATION (R): FD = {

FAC_ID →FAC_ID, FAC_NAME

DEP_ID → DEP_ID, ROOM_NO

DEP_ID → ROOM_NO

}

IN THE DEPENDACY DEP_ID → ROOM_NO, ROOM_NO IS A NON-PRIME ATTRIBUTE AND IS PARTIALLY DEPENDED ON DEP_ID

THIS VIOLATES THE 2ND NORMAL FORM RULES.

SO, WE HAVE TO DECOMPOSE THE RELATION INTO R1 AND R2

R1: (FAC_ID, FAC_NAME, DEP_ID)

FD:

{

FAC_ID → FAC_ID, FAC_NAME DEP_ID → DEP_ID

}

CK: (FAC_ID, DEP_ID) SO,

PA: (FAC_ID, DEP_ID)

NPA: (FAC_NAME)

R2:(DEP_ID, ROOM_NO)

FD:

{

DEP_ID → DEP_ID, ROOM_NO

}

CK: (DEP_ID) SO,

PA: (DEP_ID)

NPA: (ROOM_NO)

NOW BOTH R1 AND R2 SATISFIES 2NF.

NOTE: IT IS A LOSSLESS JOIN DECOMPOSITION AS THE DECOMPOSITION HAS CANDIDATE KEY OF BOTH R1 AND R2 AS ITS COMMON ATTRIBUTE

```
CREATE VIEW ROOMS AS
SELECT DISTINCT DEP_ID, ROOM_NO
 FROM FACULTY;

SELECT * FROM ROOMS;

ALTER TABLE FACULTY
DROP COLUMN ROOM_NO;
SELECT * FROM FACULTY;
```

View created.

Table altered.

DEP_ID	ROOM_NO
D03	3
D06	6
D01	1
D04	4
D05	5
D09	9
D07	7
D08	8
D02	2

Download CSV

9 rows selected.

FAC_ID	FAC_NAME	DEP_ID
F01	PREM SINGH	D01
F02	MANSA DEVI	D01
F03	BHAVANI	D02
F04	SRINIVAS RAO	D02
F05	VIJAY SINGH	D03
F06	PRAVEEN KUMAR	D03
F07	VANDANA SINGH	D04
F08	RAGHAVENDRA	D04
F09	GHANSHYAM	D05
F10	ABHISHEK GV	D05
F11	SASI KUMAR	D06
F12	L GONDI	D06
F13	DON S	D07
F14	RAVI	D07

<u>3RD NORMAL FORM</u>

1ST LET'S CHECK 3NF FOR R1 AND R2 -

R1: (FAC_ID, FAC_NAME, DEP_ID)

FD:

{

FAC_ID → FAC_ID, FAC_NAME DEP_ID → DEP_ID,

}

CK: (FAC_ID, DEP_ID) PA: (FAC_ID, DEP_ID) NPA: (FAC_NAME)

R2: (DEP_ID, ROOM_NO)

FD:

{

DEP_ID → DEP_ID, ROOM_NO

}

CK: (DEP_ID)

PA: (DEP_ID)

NPA: (ROOM_NO)

3NF IS SATISFIED FOR R1 AND R2

NOTE: IF THERE IS NO CHANCE TO FURTHER DECOMPOSE A TABLE/RELATION, THEN IT IS THE HIGHEST NORMAL FORM OF THAT TABLE.

SO, WE NEED NOT CONTINUE TO CHECK R2 AND R1 FOR FURTHER NORMALIZATION.

NORMAL FORM OR BCNF

a. The table should be in 3NF.

b. Every attribute must be dependent on a super key, i.e. all LHS values must be either candidate or super key.

SO, 1ST LET'S CHECK 3NF FOR R1 AND R2 -

R1: (FAC_ID, FAC_NAME, DEP_ID)

FD:

{

FAC_ID → FAC_ID, FAC_NAME DEP_ID → DEP_ID,

}

CK: (FAC_ID, DEP_ID) PA: (FAC_ID, DEP_ID) NPA: (FAC_NAME)

R2: (DEP_ID, ROOM_NO)

FD:

{

DEP_ID → DEP_ID, ROOM_NO

}

CK: (DEP_ID)

PA: (DEP_ID)

NPA: (ROOM_NO)

3.5NF IS SATISFIED FOR R1 AND R2

2. 4TH NORMAL FORM

a. The table should be in BCNF.

b. No multivalued dependency should be present

(x->->y)

c. For Table with ABC columns, B and C should be independent

4NF IS SATISFIED FOR R1 AND R2

3.5TH NORMAL FORM

a. The table should be in 4NF.

b. There should only be lossless join dependency.

5NF IS SATISFIED FOR R1 AND R2