SCHOOL MANAGEMENT SYSTEM

by

Rudhar Sharma – C0871405

Reea Sagar – C0869344

Fahad Ismail Mohammed – C0872742

Akhilesh K Saju – C0879963

Praveen Pallipurath – C0879905

TABLE OF CONTENTS

INTRODUCTION TO THE PROCESS:

- The purpose of this project is to demonstrate a student database management system in its simplest form that uses a digital tracking system to maintain a record of all the students. However, in practicality, the system has much more complex functions.
- It maintains a record of every student-related data ranging from fees and financial records, examination records, transport facilities being provided by the school and availed by the students along with the usage of libraries, laboratory, computer lab and other school facilities used regularly by the students.
- A student database management system allows schools to save and access these records as needed by them, thus simplifying the work of the school administration team.
- The attendance management system uses a biometric system or an access card to maintain a quick and accurate record of the student and school staff attendance. The system saves class teachers tedious minutes every morning calling out the class attendance and provides with the administration team on timely summaries and records of regular latecomers, absentees etc. when needed.

- The system also allows for attendance updates to be sent to parents. It enables schools to track the attendance of the teachers and the various staff and calculate leaves and working days with ease and accuracy.

ENTITIES AND ATTRIBUTES

An entity in DBMS (Database management System) is **a real-world thing or a real-world object which is distinguishable from other objects in the real world**. For example, a car is an entity.

The entities and attributes we used in our project:

Student:

Attributes: AddressID, First_name, Last_name, parent_name, Dob, Mobile, Email, Joining_date.

Primary Key for student is Student_ID.

Foreign Key for student is Address_ID.

student	
PK	student_ID
FK	address_id
	first_name
	last_name
	guardian_name
	dob
	mobile
	email
	joining_date

Staff:

Attributes: Staff_Id, First_name, Last_name, Phone, Email, Dob, Designation.

Primary key is Staff_ID.

Staff	
PK	staff_ID
	first_name
	last_name
	phone
	email
	dob
	designation

Courses:

Attributes: Course_ID, Grade_ID, name, Description, Credits.

Primary key for Courses is Course_ID.

Foreign Key for Courses is Grade_ID.

course	
PK	course_id
FK	grade_id
	name
	description
	credits

Class:

Attributes: Class_ID, Grade_Id, Staff_id, Year.

Primary key for Class Is Class_ID.

Foreign key for Class Is Grade_ID.

Foreign key for Class Is Staff_ID.

class	
PK	class_enrolled_id
FK	grade_id
FK	staff_id
	year_of_attempt

Exam:

Attributes: Exam_Id, Course_id, Exam_type_id, Student_id, Start_date.

Primary Key is Exam_ID.

Foreign Key is Exam_type_ID.

Foreign Key is student_ID.

Foreign Key is Course_ID.

exam	
PK	exam_val_id
FK	course_id
FK	exam_type_id
FK	student_id
	start_date

Exam result:

Attributes: Exam_id, Course_id, Marks.

Primary key for Exam result is Exam_ID.

Foreign key for Exam result is Student_ID.

Foreign key for Exam result is Course_ID.

exam_result	
PK	exam_result_id
FK	student_id
FK	course_id
	marks

Exam Type:

Attributes: Exam_type_id, Type, Desc.

Primary key for Exam type is Exam_type_id.

exam_type	
PK	exam_type_id
	type
	desc

Student Council:

Attributes: Student_council_ID, Student_ID, Designation.

Primary key for Student Council is Student_council_id.

Foreign key for Student Council is Student_ id.

student_council	
PK	student_council_ID
FK	student_ID
	designation

Attendance:

Attributes: Attendance_Id, Student-Id, Date, Status.

Primary key for attendance is attendance_id.

Foreign key for attendance is Student_id.

attendance	
PK	attendance_id
FK	student_id
	current_date
	status

Grades:

Attributes: Grade_id, Name, Desc.

Primary key for Grades is grade_id.

grade	
PK	grade_id
	name
	desc

Library:

Attributes: Library_id, Student_id, Status.

Primary key for Library is Library_id.

Foreign key for Library is Student_id.

Foreign key for Library is Book_id.

library	
PK	library_id
FK	student_id
FK	book_id
	status

Book:

Attributes: Book_id, Name, Desc.

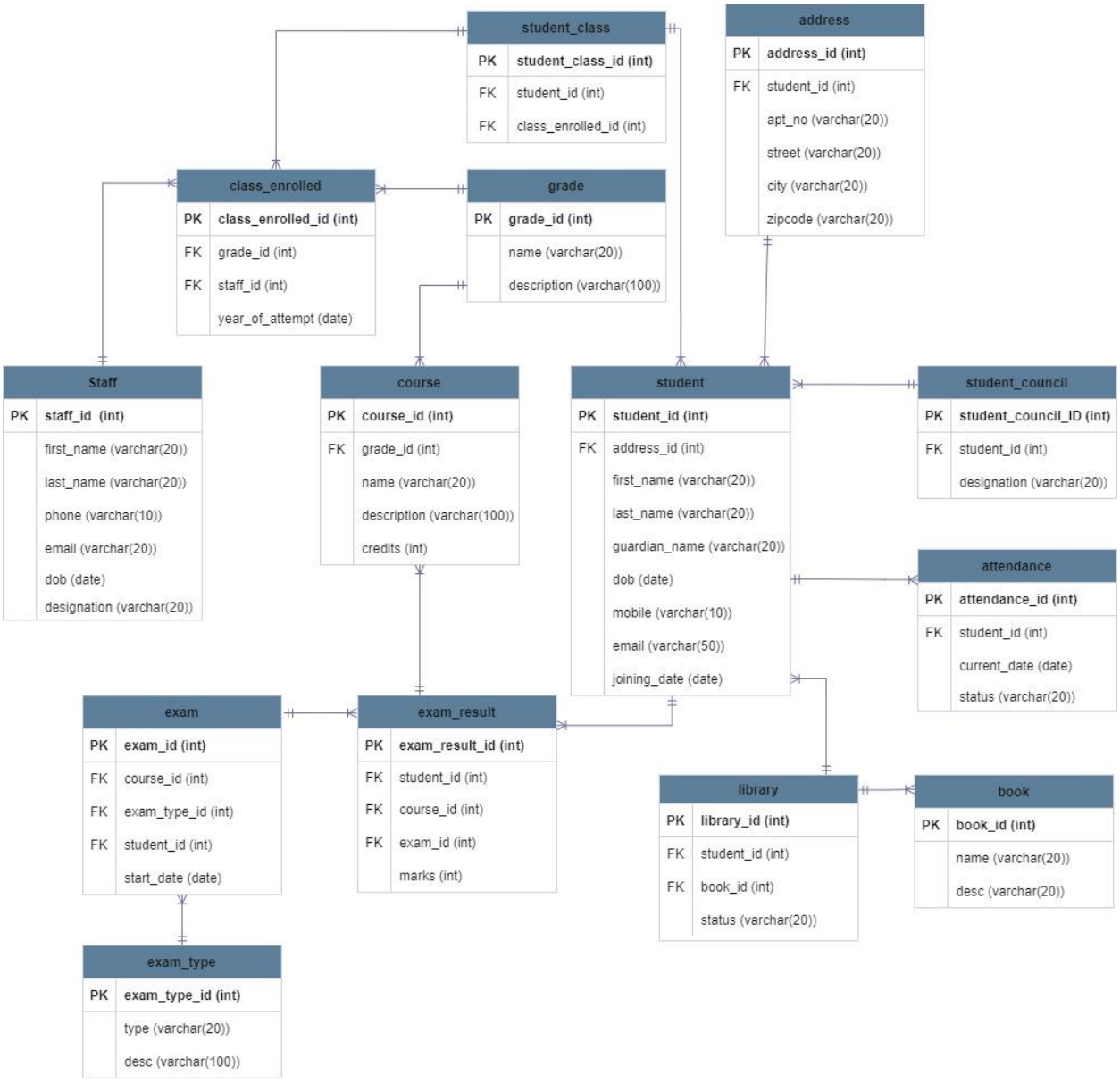
Primary key for Book is Book_id.

book	
PK	book_id
	name
	desc

Mandatory attributes: Book_ID and Name

Optional Attributes: Desc

Entity Relational Diagram:



2. Normalization: It is a database design technique that reduces data redundancy and eliminates undesirable characteristics like Insertion, Update and Deletion Anomalies. Normalization rules divides larger tables into smaller tables and links them using relationships. The purpose of Normalization in SQL is to eliminate redundant (repetitive) data and ensure data is stored logically.

- Criteria for 1NF, 2NF and 3NF
  - 1NF :
    - Each table cell should contain a SINGLE VALUE.
    - Each record needs to be UNIQUE
  - 2NF :
    - Rule-1 Be in 1NF.
    - Rule-2 Single Column Primary Key that does not functionally dependent on any subset of candidate key relation.
  - 3NF :
    - Rule-1 Be in 2NF.
      - Rule-2 has no transitive functional dependencies.

- In all the tables (*Student, Class, Course, Grade, Student_Class, Address, Staff, Student_Council, Attendance, Exam, Exam_result, Exam_Type, Library and Book*).

  - **1NF:**

✓ Primary Key is defined.
✓ No multivalued Columns and repeating groups found.
✓ Hence, they are in 1NF.

  - **2NF:**

✓ No partial dependencies found in all the tables.
✓ Hence, they are in 2NF.

  - **3NF:**

✓ No transitive dependencies found in all the tables.
✓ Hence, they are in 3NF.

SQL QUERIES FOR DATA INSERTION AND MODIFICATION

```
SELECT * FROM STUDENT;

SELECT * FROM ADRESS;

SELECT * FROM STUDENT_COUNCIL;

SELECT * FROM ATTENDANCE;

SELECT * FROM BOOK;

SELECT * FROM LIBRARY;

SELECT * FROM EXAM_TYPE;

SELECT * FROM STAFF;

SELECT * FROM GRADE;

SELECT * FROM STUDENT_CLASS;

SELECT * FROM CLASS_ENROLLED;

SELECT * FROM COURSE;

SELECT * FROM EXAM_RESULT;

SELECT * FROM EXAM;


INSERT INTO "STUDENT" (STUDENT_ID, ADDRESS_ID, FIRST_NAME, LAST_NAME, GUARDIAN_NAME,
DOB, MOBILE, EMAIL, JOINING_DATE)

VALUES ('1', '1', 'akhilesh', 'saju', 'saju', TO_DATE('1997-12-05 12:09:45', 'YYYY-MM-DD HH24:MI:SS'),
'6478325376', 'akhileshsaju@gmail.com', TO_DATE('2022-12-06 12:10:17', 'YYYY-MM-DD
HH24:MI:SS'))


INSERT INTO "ADRESS" (
 address_id,
 apt_no,
 street,
 city,
 zipcode
)
```

```
VALUES (1, '14', 'Mclaughin', 'Brmapton', 'L5X6W1')
```

```
INSERT INTO "STUDENT" (STUDENT_ID, ADDRESS_ID, FIRST_NAME, LAST_NAME, GUARDIAN_NAME,
DOB, MOBILE, EMAIL, JOINING_DATE)

VALUES ('2', '2', 'praveen', 'saji', 'saji', TO_DATE('1997-06-03 12:11:02', 'YYYY-MM-DD HH24:MI:SS'),
'6475687689', 'praveen@hotmail.com', TO_DATE('2022-12-06 15:10:17', 'YYYY-MM-DD HH24:MI:SS'))
```

```
INSERT INTO "ADRESS" (
 address_id,
 apt_no,
 street,
 city,
 zipcode
)
VALUES (2, '69', 'Williams Parkway', 'Brmapton', 'L3X2C1')
```

```
create table Student_Council(
student_council_id int primary key,
student_id int,
Designation varchar(20),
CONSTRAINT fk_student
   FOREIGN KEY (Student_id)
   REFERENCES student(Student_id)
)
```

```
INSERT INTO "STUDENT_COUNCIL" (STUDENT_COUNCIL_ID, STUDENT_ID, DESIGNATION) VALUES ('1', '1', 'm')

INSERT INTO "STUDENT_COUNCIL" (STUDENT_COUNCIL_ID, STUDENT_ID, DESIGNATION) VALUES ('2', '2', 'p')
```

```
create table Attendance(

Attendance_id int primary key,

Student_id int,

Current_Date date not null,

Status varchar(20) CONSTRAINT check_status_of_attendance

        CHECK (Status in ('present','absent'))disable,

CONSTRAINT fk_student_check

   FOREIGN KEY (Student_id)

   REFERENCES student(Student_id)

)
```

```
INSERT INTO "ATTENDANCE" (ATTENDANCE_ID, STUDENT_ID, CURRENT_DATE, STATUS) VALUES ('1', '2', TO_DATE('2022-12-06 12:19:02', 'YYYY-MM-DD HH24:MI:SS'), 'p')

INSERT INTO "ATTENDANCE" (ATTENDANCE_ID, STUDENT_ID, CURRENT_DATE, STATUS) VALUES ('2', '1', TO_DATE('2022-12-06 12:20:42', 'YYYY-MM-DD HH24:MI:SS'), 'np')
```

```
create table book(

book_id int primary key,
```

```
name varchar(20),
description varchar(20))

INSERT INTO book(
book_id,
name,
description)
VALUES(1, 'Intro to python', 'A book for python')

INSERT INTO book(
book_id,
name,
description)
VALUES(2, 'Intro to Data', 'Data concepts')


create table library(
library_id int primary key,
student_id int,
book_id int,
status varchar(20),
CONSTRAINT fk_student_check1
   FOREIGN KEY (student_id)
   REFERENCES student(Student_id),
CONSTRAINT fk_book_check1
   FOREIGN KEY (book_id)
   REFERENCES Book(book_id)

)
```

```
INSERT INTO library(
library_id,
student_id,
book_id,
status
)
VALUES (11, 1, 2, 'Given')
VALUES (12, 2, 1, 'Returned')
```

```
create table exam_type(
exam_type_id int primary key,
type varchar(20),
description varchar(100))
```

```
INSERT INTO "EXAM_TYPE" (EXAM_TYPE_ID, TYPE, DESCRIPTION) VALUES ('1', 'mid term', 'first exam')
INSERT INTO "EXAM_TYPE" (EXAM_TYPE_ID, TYPE, DESCRIPTION) VALUES ('2', 'last tern', 'final exam')
```

```
create table staff(
staff_id int primary key,
first_name varchar(20) not null,
last_name varchar(20),
phone varchar(10),
email varchar(30),
```

```
dob date not null,
designation varchar(20)
)

INSERT INTO "STAFF" (STAFF_ID, FIRST_NAME, LAST_NAME, PHONE, EMAIL, DOB, DESIGNATION)
VALUES ('1', 'aman', 'kaul', '6478324567', 'aman@gmail', TO_DATE('1996-10-04 12:29:17', 'YYYY-MM-DD HH24:MI:SS'), 'teacher')

INSERT INTO "STAFF" (STAFF_ID, FIRST_NAME, LAST_NAME, PHONE, EMAIL, DOB, DESIGNATION)
VALUES ('2', 'athul', 'madav', '6478354672', 'athul123@mail.com', TO_DATE('1995-12-07 12:30:28', 'YYYY-MM-DD HH24:MI:SS'), 'teacher')

INSERT INTO "STAFF" (STAFF_ID, FIRST_NAME, LAST_NAME, PHONE, EMAIL, DOB, DESIGNATION)
VALUES ('3', 'hari', 'mohan', '5678324567', 'harimohan@gmail.com', TO_DATE('1998-08-11 12:31:35', 'YYYY-MM-DD HH24:MI:SS'), 'head of department')

create table grade(
grade_id int primary key,
name varchar(20),
description varchar(100))

INSERT INTO "GRADE" (GRADE_ID, NAME, DESCRIPTION) VALUES ('1', 'th grade', 'high school')
INSERT INTO "GRADE" (GRADE_ID, NAME, DESCRIPTION) VALUES ('2', '9th grade', 'high school')

create table student_class(
class_id int primary key,
student_id int,
constraint check_student_class
 foreign key(student_id)
```

```
 references student(Student_id)
)

INSERT INTO "STUDENT_CLASS" (CLASS_ID, STUDENT_ID) VALUES ('1', '1')
INSERT INTO "STUDENT_CLASS" (CLASS_ID, STUDENT_ID) VALUES ('2', '2')

Create table class_enrolled(
class_enrolled_id  int primary key,
grade_id int,
staff_id int,
year_of_attempt date,
constraint check_grade_class
 foreign key(grade_id)
 references grade(grade_id),
constraint check_staff_class
 foreign key(staff_id)
 references staff(staff_id)
)

INSERT INTO "CLASS_ENROLLED" (CLASS_ENROLLED_ID, GRADE_ID, STAFF_ID, YEAR_OF_ATTEMPT)
VALUES ('1', '1', '1', TO_DATE('2022-12-06 12:35:33', 'YYYY-MM-DD HH24:MI:SS'))
INSERT INTO "CLASS_ENROLLED" (CLASS_ENROLLED_ID, GRADE_ID, STAFF_ID, YEAR_OF_ATTEMPT)
VALUES ('2', '2', '2',TO_DATE('2022-12-06 12:35:38', 'YYYY-MM-DD HH24:MI:SS'))

create table course(
```

```
course_id int primary key,
grade_id int,
name varchar(20),
description varchar(100),
credits int,
constraint check_grade
foreign key(grade_id)
references grade(grade_id)
)

INSERT INTO "COURSE" (COURSE_ID, GRADE_ID, NAME, DESCRIPTION, CREDITS) VALUES ('1', '1', 'data base', 'basic sql queries', '10')
INSERT INTO "COURSE" (COURSE_ID, GRADE_ID, NAME, DESCRIPTION, CREDITS) VALUES ('2', '2', 'python', 'basic python gaming', '10')


create table exam_result(
exam_result_id int primary key,
student_id int,
course_id int,
marks int,
constraint check_course
foreign key(course_id)
references course(course_id),
constraint check_student_exam_result
foreign key(student_id)
references student(student_id)
)
```

```
INSERT INTO "EXAM_RESULT" (EXAM_RESULT_ID, STUDENT_ID, COURSE_ID, MARKS) VALUES ('1', '1', '1', '10')

INSERT INTO "EXAM_RESULT" (EXAM_RESULT_ID, STUDENT_ID, COURSE_ID, MARKS) VALUES ('2', '1', '2', '9')



create table exam (

exam_val_id int primary key,

course_id int,

exam_type_id int,

student_id int ,

start_date date not null,

constraint check_student_exam

foreign key(student_id)

references student(student_id),

constraint check_course_exam

foreign key(course_id)

references course(course_id),

constraint check_exam_type_id

foreign key(exam_type_id)

references exam_type(exam_type_id)

)


INSERT INTO "EXAM" (EXAM_VAL_ID, COURSE_ID, EXAM_TYPE_ID, STUDENT_ID, START_DATE) VALUES ('1', '1', '1', '1', TO_DATE('2022-12-08 12:50:24', 'YYYY-MM-DD HH24:MI:SS'))

INSERT INTO "EXAM" (EXAM_VAL_ID, COURSE_ID, EXAM_TYPE_ID, STUDENT_ID, START_DATE) VALUES ('2', '2', '1', '1', TO_DATE('2022-12-09 12:50:53', 'YYYY-MM-DD HH24:MI:SS'))

INSERT INTO "EXAM" (EXAM_VAL_ID, COURSE_ID, EXAM_TYPE_ID, STUDENT_ID, START_DATE) VALUES ('3', '1', '1', '2', TO_DATE('2022-12-08 12:51:38', 'YYYY-MM-DD HH24:MI:SS'))

INSERT INTO "EXAM" (EXAM_VAL_ID, COURSE_ID, EXAM_TYPE_ID, STUDENT_ID, START_DATE) VALUES ('4', '2', '1', '2', TO_DATE('2022-12-09 12:52:08', 'YYYY-MM-DD HH24:MI:SS'))
```

NORMALIZED TABLES

Student & Address

STUDENT_ID	ADDRESS_ID	FIRST_NAME	LAST_NAME	GUARDIAN_NAME	DOB	MOBILE	EMAIL	JOINING_DATE
1	1	akhilesh	saju	saju	05-DEC-97	6478325376	akhileshsaju@gmail.com	06-DEC-22
2	2	praveen	saji	saji	03-JUN-97	6475687689	praveen@hotmail.com	06-DEC-22

Download CSV
2 rows selected.

ADDRESS_ID	APT_NO	STREET	CITY	ZIPCODE
2	69	Williams Parkway	Brmapton	L3X2C1
1	14	Mclaughin	Brmapton	L5X6W1

- **1NF:**

✓ Primary Key is defined.
✓ No multivalued Columns and repeating groups found.
✓ Hence, they are in 1NF.

- **2NF:**

✓ No partial dependencies found in all the tables.
✓ Hence, they are in 2NF.

- **3NF:**

✓ No transitive dependencies found in all the tables.
✓ Hence, they are in 3NF.

Student_Council

STUDENT_COUNCIL_ID	STUDENT_ID	DESIGNATION
1	1	m
2	2	p

- **1NF:**

✓ Primary Key is defined.
✓ No multivalued Columns and repeating groups found.

✓ Hence, they are in 1NF.

- **2NF:**

✓ No partial dependencies found in all the tables.
✓ Hence, they are in 2NF.

- **3NF:**

✓ No transitive dependencies found in all the tables.
✓ Hence, they are in 3NF.

Attendance

ATTENDANCE_ID	STUDENT_ID	CURRENT_DATE	STATUS
1	2	06-DEC-22	p
2	1	06-DEC-22	np

- **1NF:**

✓ Primary Key is defined.
✓ No multivalued Columns and repeating groups found.
✓ Hence, they are in 1NF.

- **2NF:**

✓ No partial dependencies found in all the tables.
✓ Hence, they are in 2NF.

- **3NF:**

✓ No transitive dependencies found in all the tables.
✓ Hence, they are in 3NF.

Book

BOOK_ID	NAME	DESCRIPTION
1	Intro to python	A book for python
2	Intro to Data	Data concepts

- **1NF:**

✓ Primary Key is defined.
✓ No multivalued Columns and repeating groups found.
✓ Hence, they are in 1NF.

- **2NF:**

✓ No partial dependencies found in all the tables.
✓ Hence, they are in 2NF.

- **3NF:**

✓ No transitive dependencies found in all the tables.
✓ Hence, they are in 3NF.

Library

LIBRARY_ID	STUDENT_ID	BOOK_ID	STATUS
11	1	2	Given
12	2	1	Returned

- **1NF:**

✓ Primary Key is defined.
✓ No multivalued Columns and repeating groups found.
✓ Hence, they are in 1NF.

- **2NF:**

✓ No partial dependencies found in all the tables.
✓ Hence, they are in 2NF.

- **3NF:**

✓ No transitive dependencies found in all the tables.
✓ Hence, they are in 3NF.

Exam_Type

EXAM_TYPE_ID	TYPE	DESCRIPTION
1	mid term	first exam
2	last tern	final exam

- **1NF:**

✓ Primary Key is defined.
✓ No multivalued Columns and repeating groups found.
✓ Hence, they are in 1NF.

- **2NF:**

✓ No partial dependencies found in all the tables.
✓ Hence, they are in 2NF.

- **3NF:**

✓ No transitive dependencies found in all the tables.
✓ Hence, they are in 3NF.

Staff

STAFF_ID	FIRST_NAME	LAST_NAME	PHONE	EMAIL	DOB	DESIGNATION
2	athul	madav	6478354672	athul123@mail.com	07-DEC-95	teacher
1	aman	kaul	6478324567	aman@gmail	04-OCT-96	teacher
3	hari	mohan	5678324567	harimohan@gmail.com	11-AUG-98	head of department

- **1NF:**

✓ Primary Key is defined.
✓ No multivalued Columns and repeating groups found.
✓ Hence, they are in 1NF.

- **2NF:**

✓ No partial dependencies found in all the tables.
✓ Hence, they are in 2NF.

- **3NF:**

✓ No transitive dependencies found in all the tables.
✓ Hence, they are in 3NF.

Grade

GRADE_ID	NAME	DESCRIPTION
1	th grade	high school
2	9th grade	high school

- **1NF:**

✓ Primary Key is defined.
✓ No multivalued Columns and repeating groups found.
✓ Hence, they are in 1NF.

- **2NF:**

✓ No partial dependencies found in all the tables.
✓ Hence, they are in 2NF.

- **3NF:**

✓ No transitive dependencies found in all the tables.
✓ Hence, they are in 3NF.

Class

CLASS_ID	STUDENT_ID
1	1
2	2

- **1NF:**

✓ Primary Key is defined.
✓ No multivalued Columns and repeating groups found.
✓ Hence, they are in 1NF.

- **2NF:**

✓ No partial dependencies found in all the tables.
✓ Hence, they are in 2NF.

- **3NF:**

✓ No transitive dependencies found in all the tables.
✓ Hence, they are in 3NF.

Class_Enrolled

CLASS_ENROLLED_ID	GRADE_ID	STAFF_ID	YEAR_OF_ATTEMPT
1	1	1	06-DEC-22
2	2	2	06-DEC-22

- **1NF:**

✓ Primary Key is defined.
✓ No multivalued Columns and repeating groups found.
✓ Hence, they are in 1NF.

- **2NF:**

✓ No partial dependencies found in all the tables.
✓ Hence, they are in 2NF.

- **3NF:**

✓ No transitive dependencies found in all the tables.
✓ Hence, they are in 3NF.

Course

COURSE_ID	GRADE_ID	NAME	DESCRIPTION	CREDITS
1	1	data base	basic sql queries	10
2	2	python	basic python gaming	10

- **1NF:**

✓ Primary Key is defined.
✓ No multivalued Columns and repeating groups found.
✓ Hence, they are in 1NF.

- **2NF:**

✓ No partial dependencies found in all the tables.
✓ Hence, they are in 2NF.

- **3NF:**

✓ No transitive dependencies found in all the tables.
✓ Hence, they are in 3NF.

Exam_Result

EXAM_RESULT_ID	STUDENT_ID	COURSE_ID	MARKS
1	1	1	10
2	1	2	9

- **1NF:**

✓ Primary Key is defined.
✓ No multivalued Columns and repeating groups found.
✓ Hence, they are in 1NF.

- **2NF:**

✓ No partial dependencies found in all the tables.
✓ Hence, they are in 2NF.

- **3NF:**

✓ No transitive dependencies found in all the tables.
✓ Hence, they are in 3NF.

Exam_Val

EXAM_VAL_ID	COURSE_ID	EXAM_TYPE_ID	STUDENT_ID	START_DATE
3	1	1	2	08-DEC-22
1	1	1	1	08-DEC-22
2	2	1	1	09-DEC-22
4	2	1	2	09-DEC-22

- **1NF:**

✓ Primary Key is defined.
✓ No multivalued Columns and repeating groups found.
✓ Hence, they are in 1NF.

- **2NF:**

✓ No partial dependencies found in all the tables.
✓ Hence, they are in 2NF.

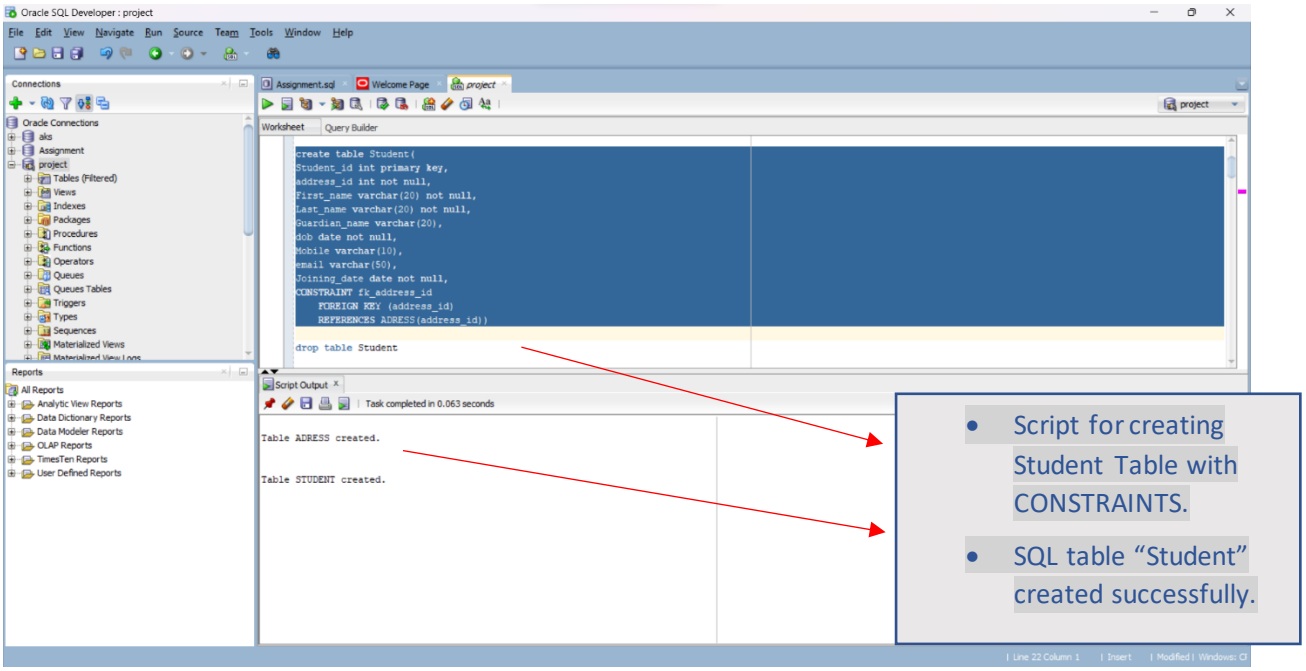
- **3NF:**

✓ No transitive dependencies found in all the tables.
✓ Hence, they are in 3NF.

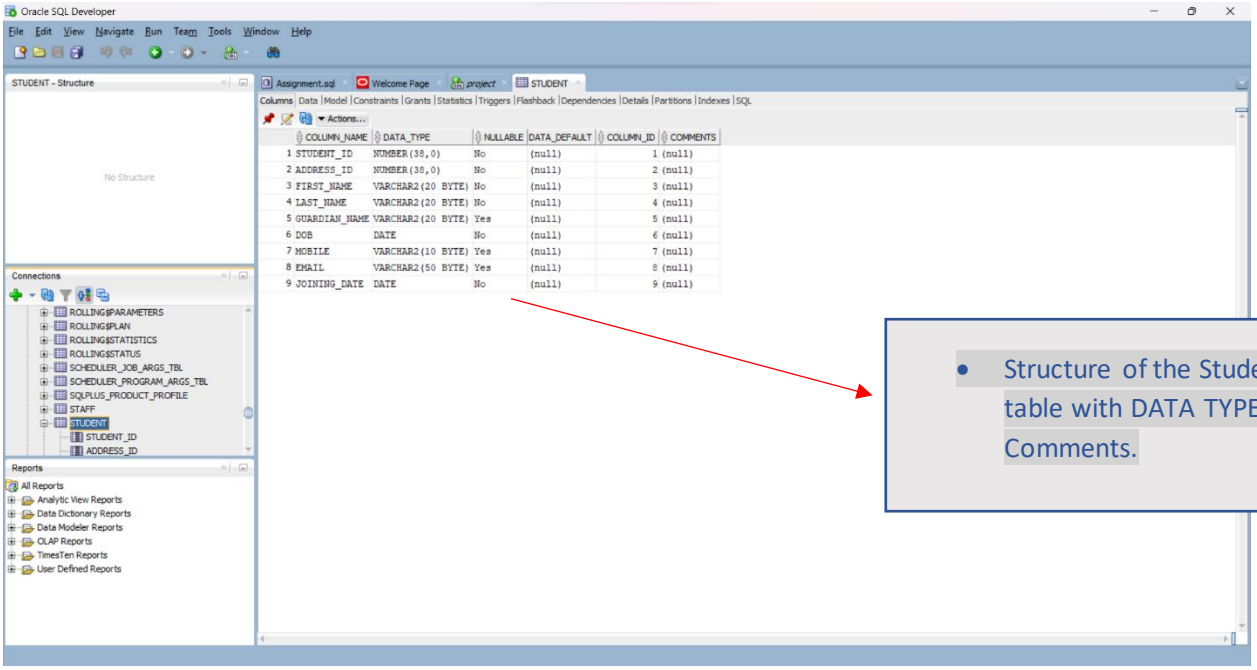
PHYSICAL DATABASE DESIGN:

Creating the table:

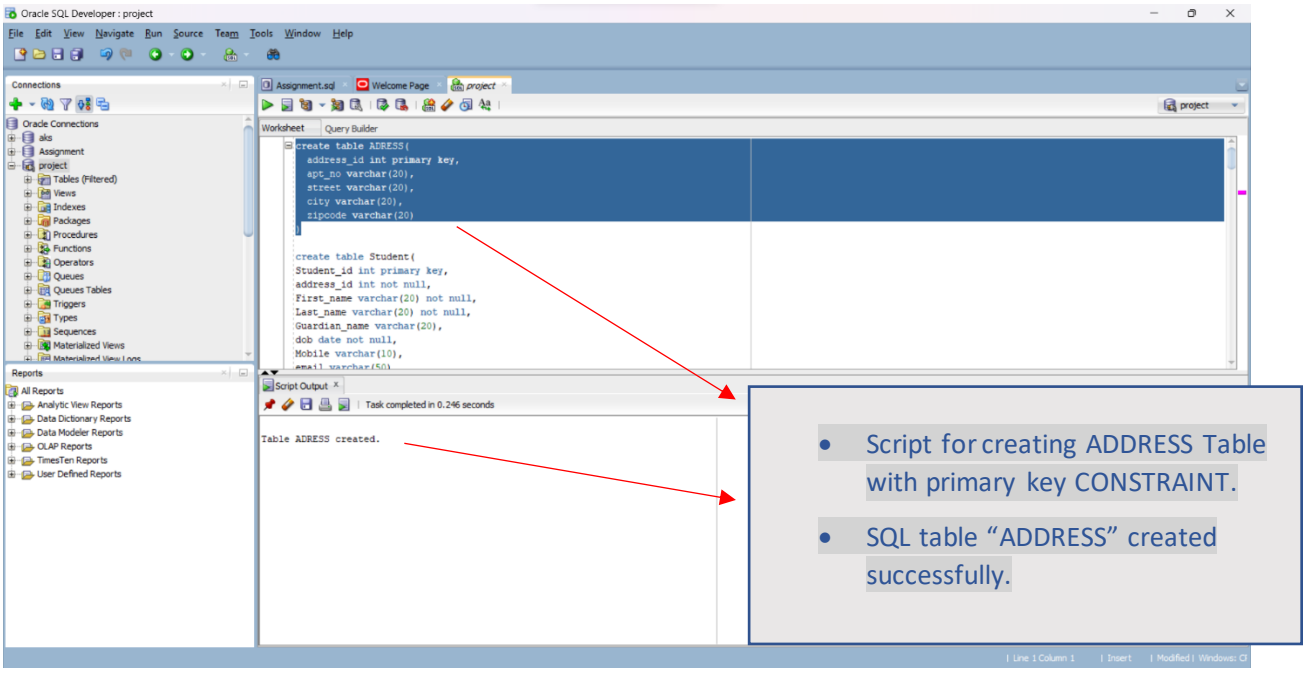
Student Table created.

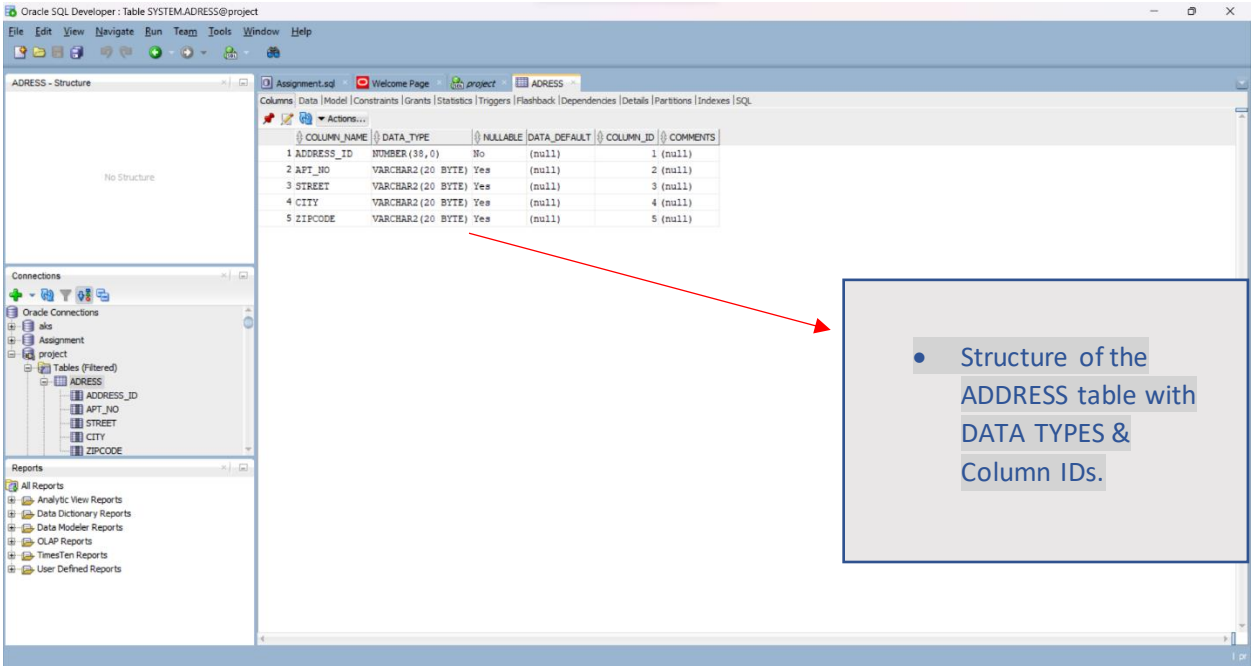


Structure of the student table.

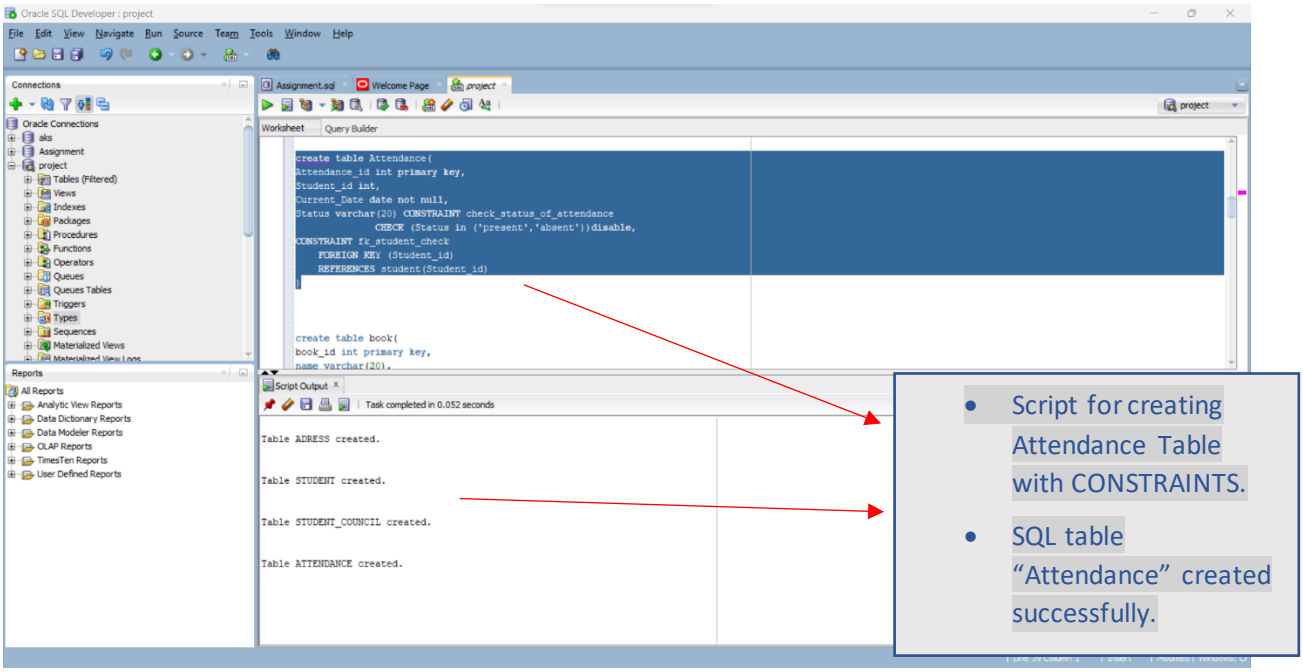


Address Table created:

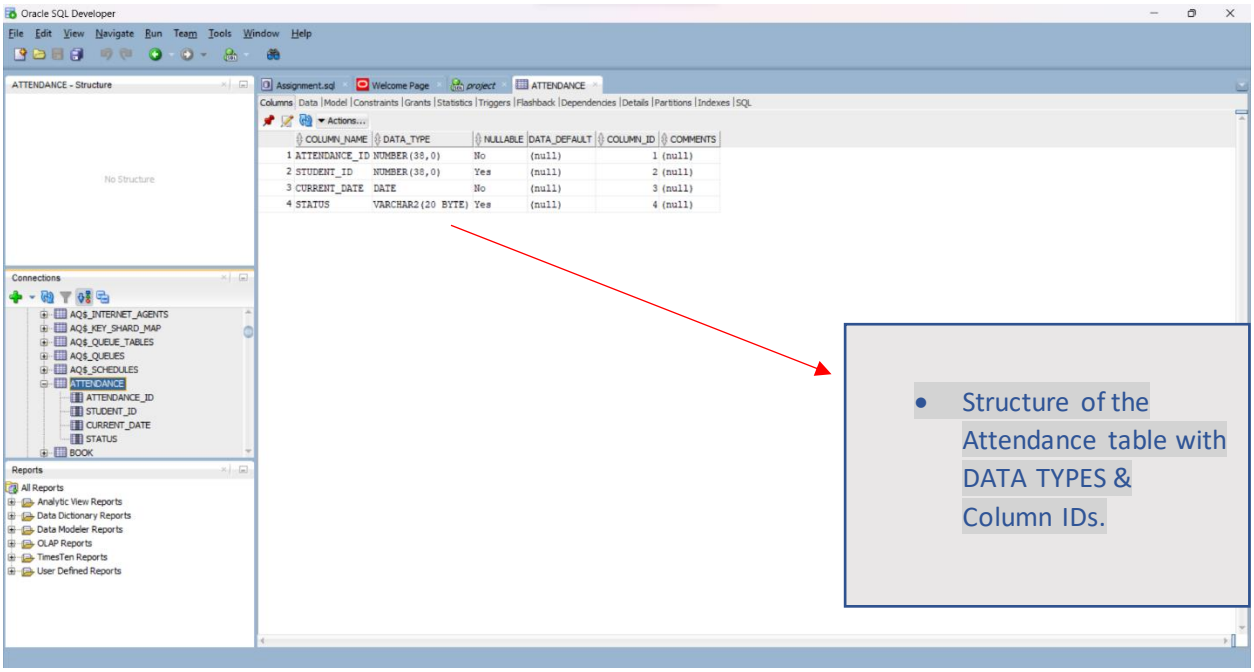




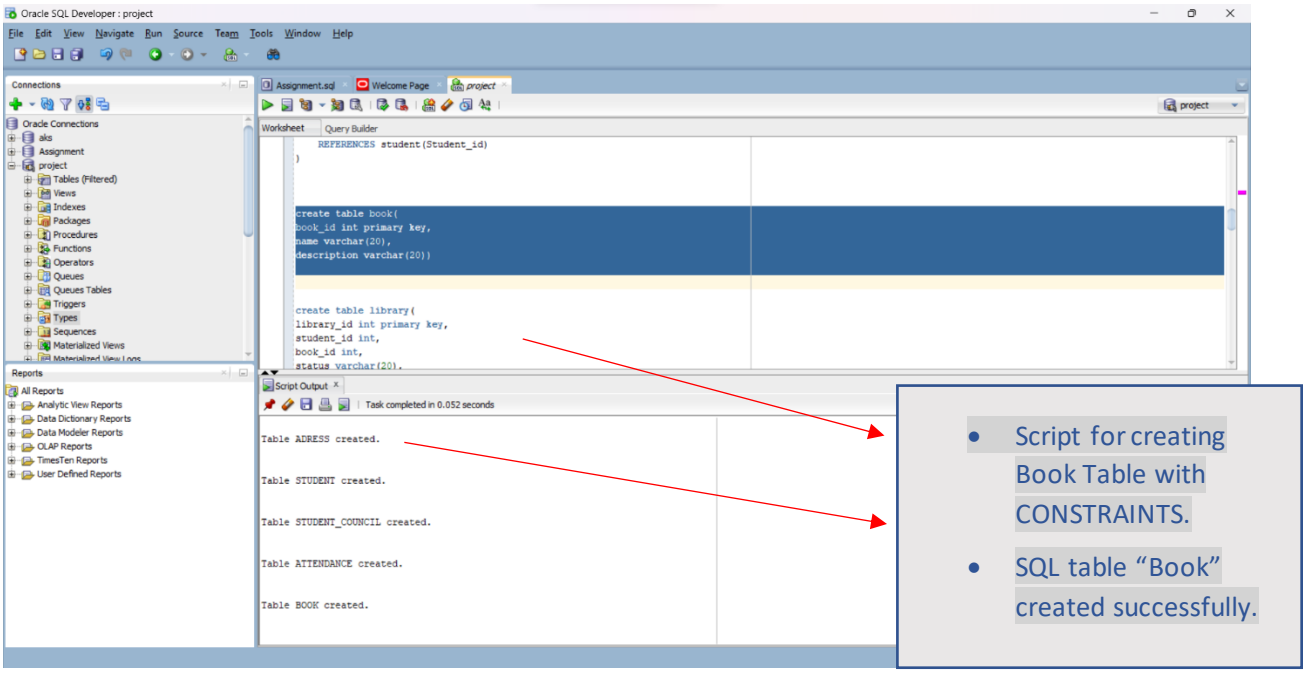
Attendance table created:



Attendance table structure.



Book table created:



Structure for book table:

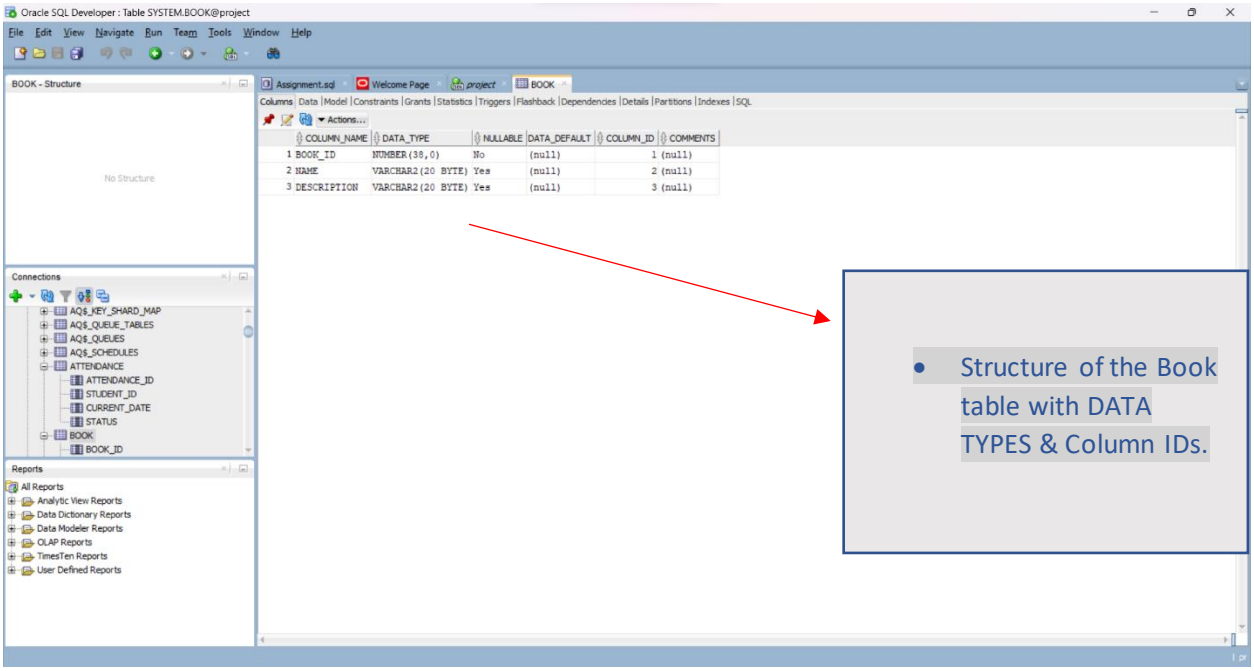
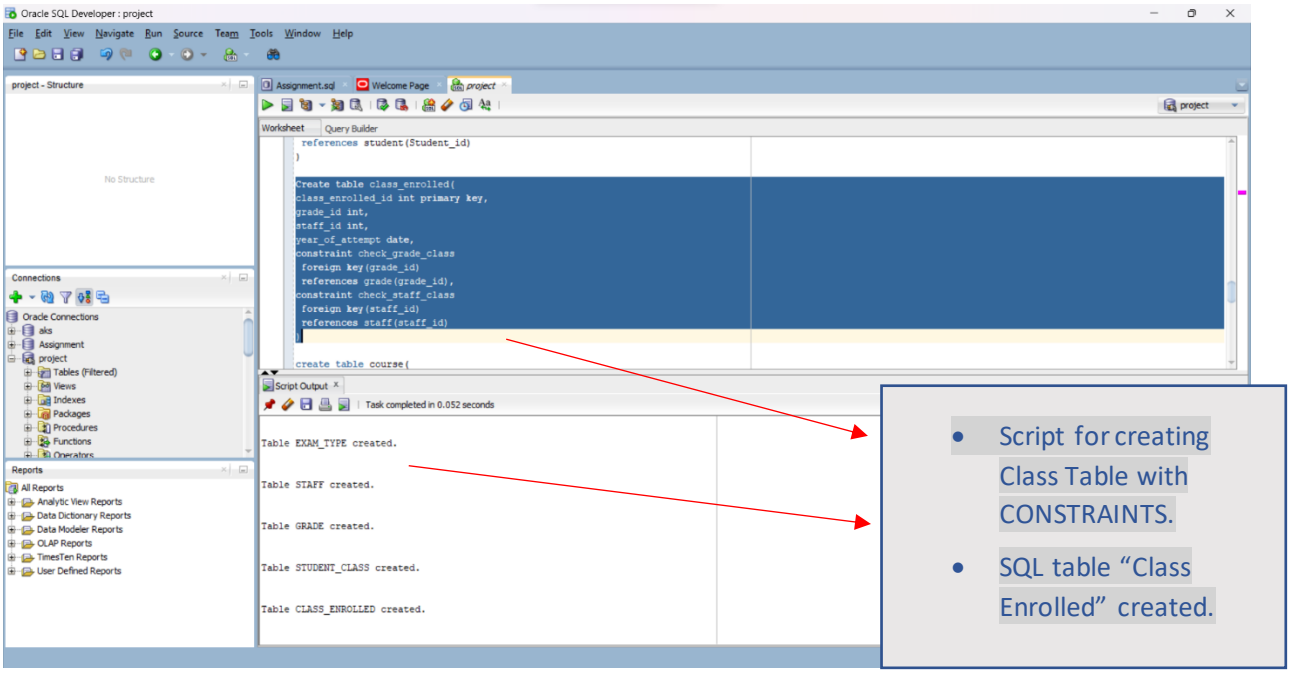
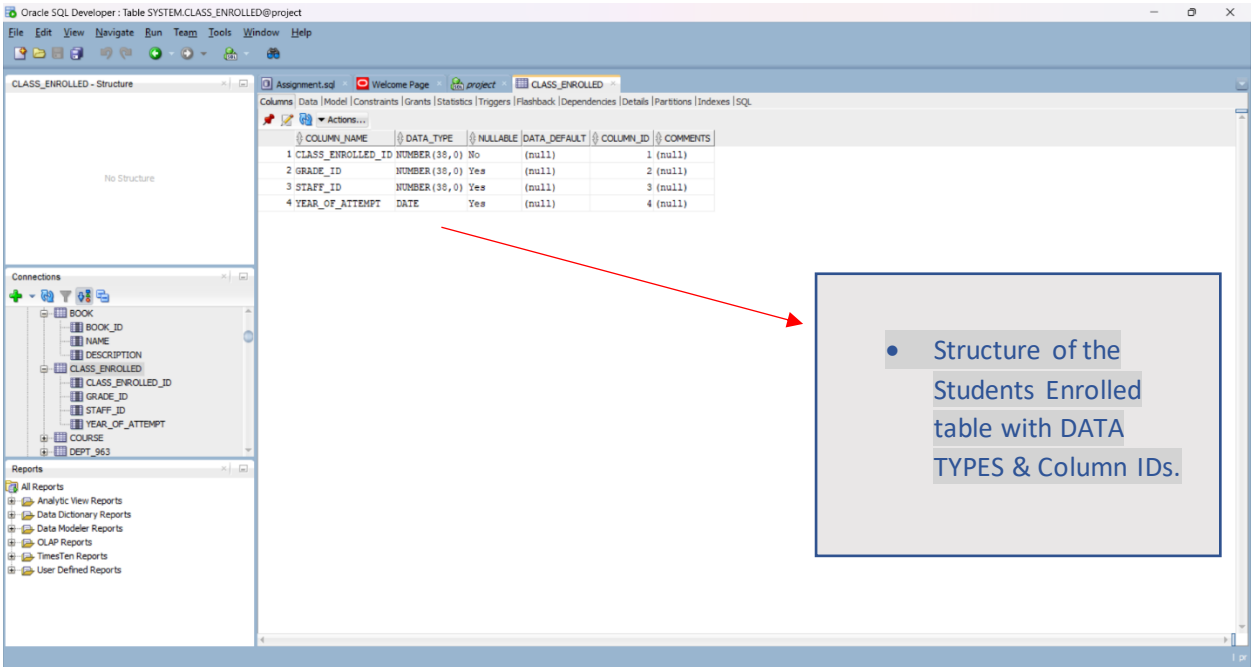


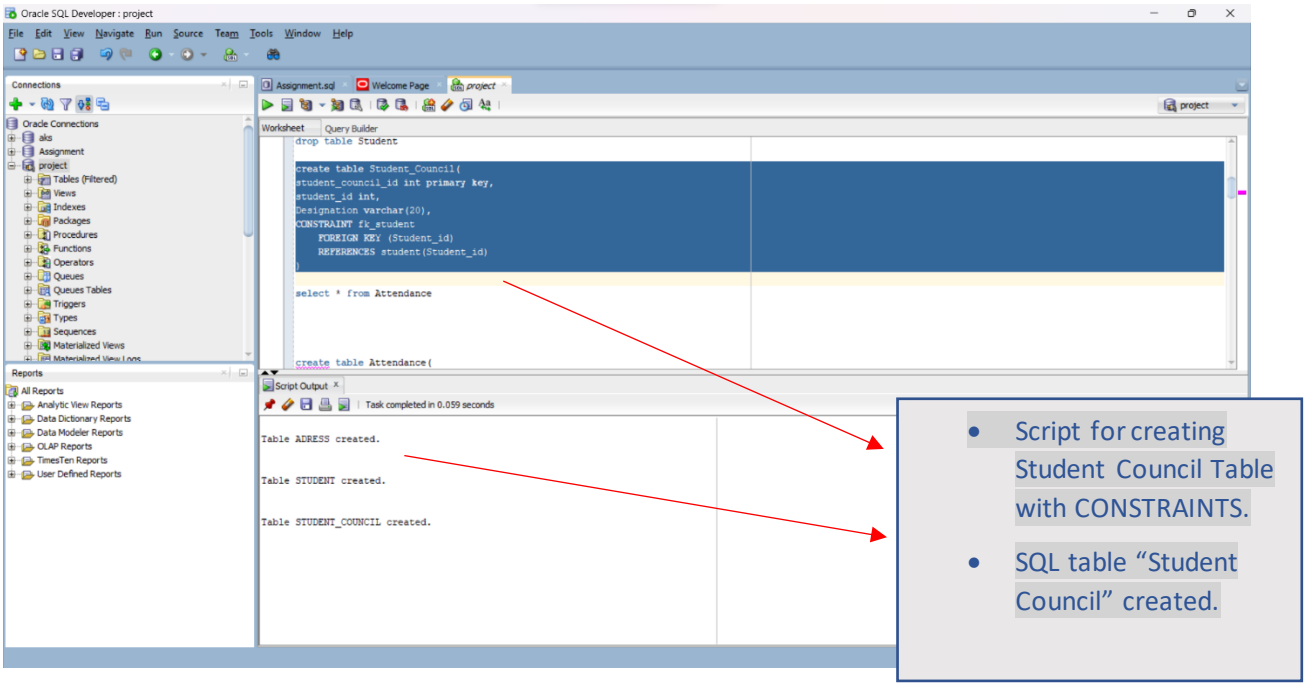
Table created for class enrolled:



Structure of students enrolled:



Student council table created:



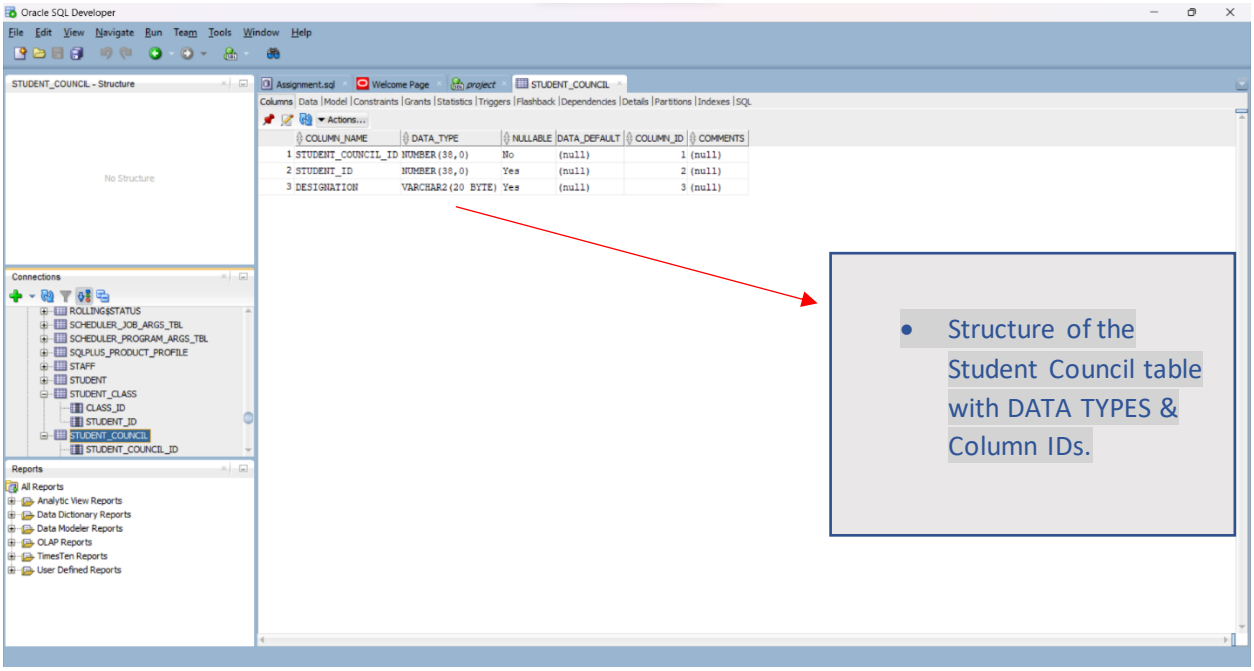
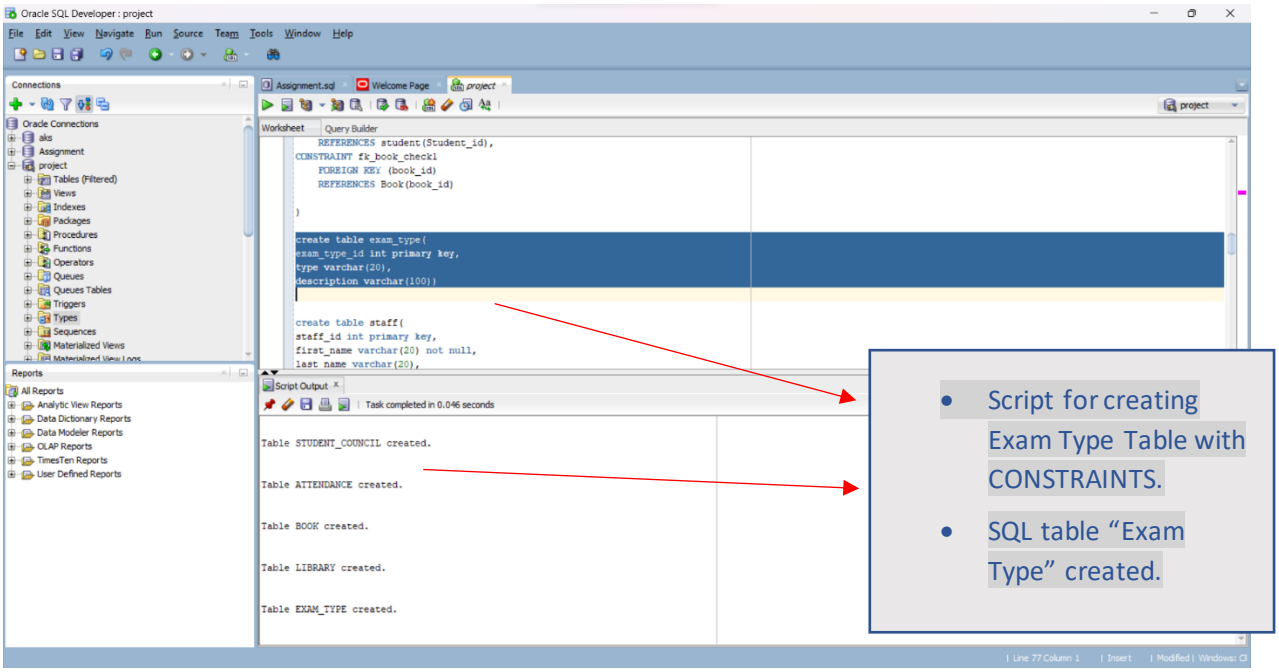


Table created for Exam Type:



Structure for exam type:

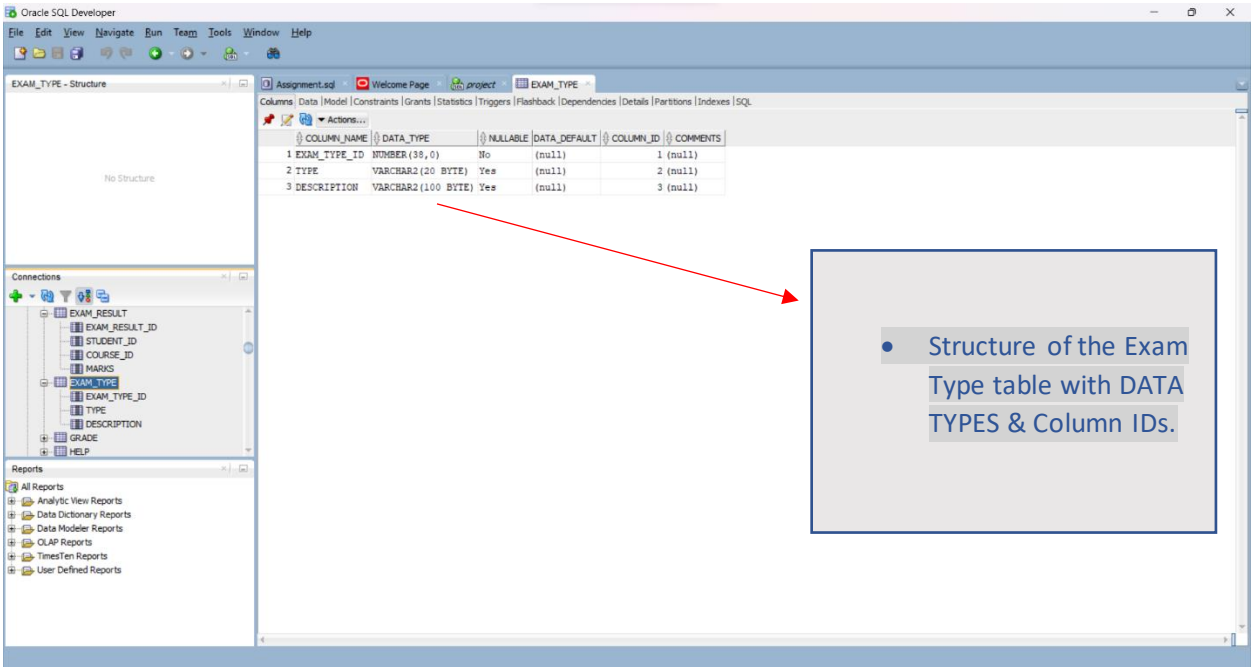


Table created for Exam:

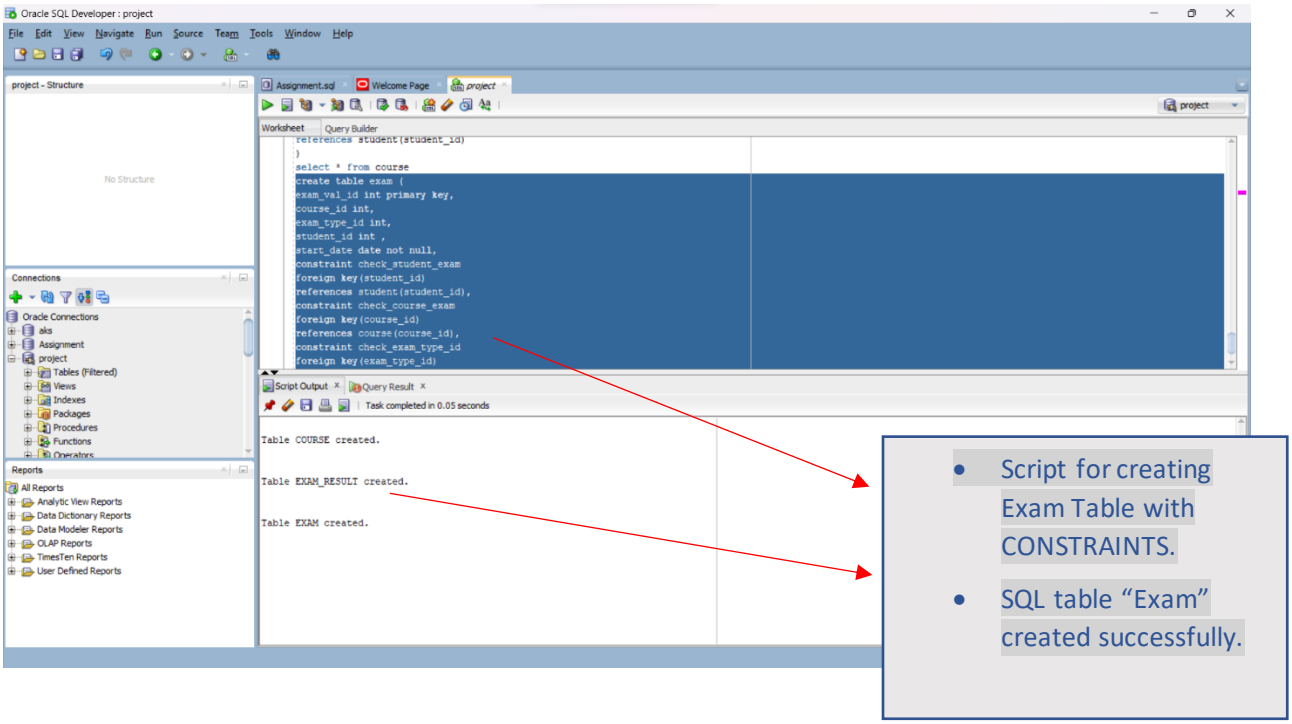
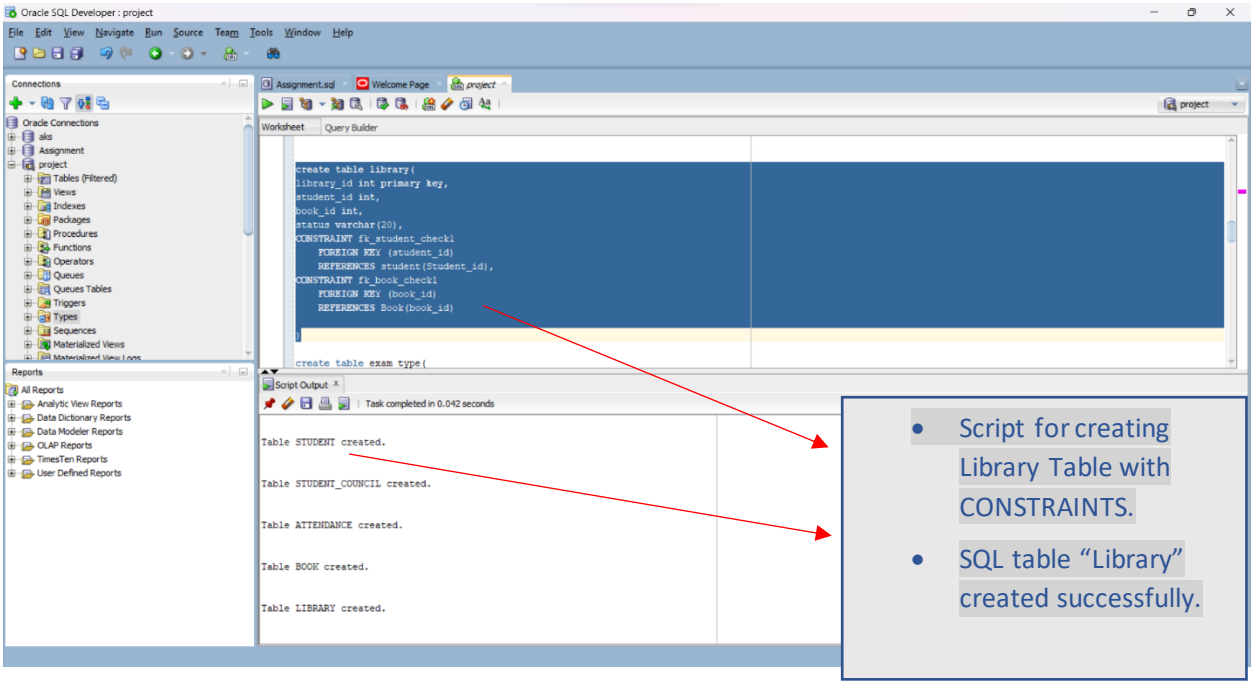


Table for library:



Structure for Library:

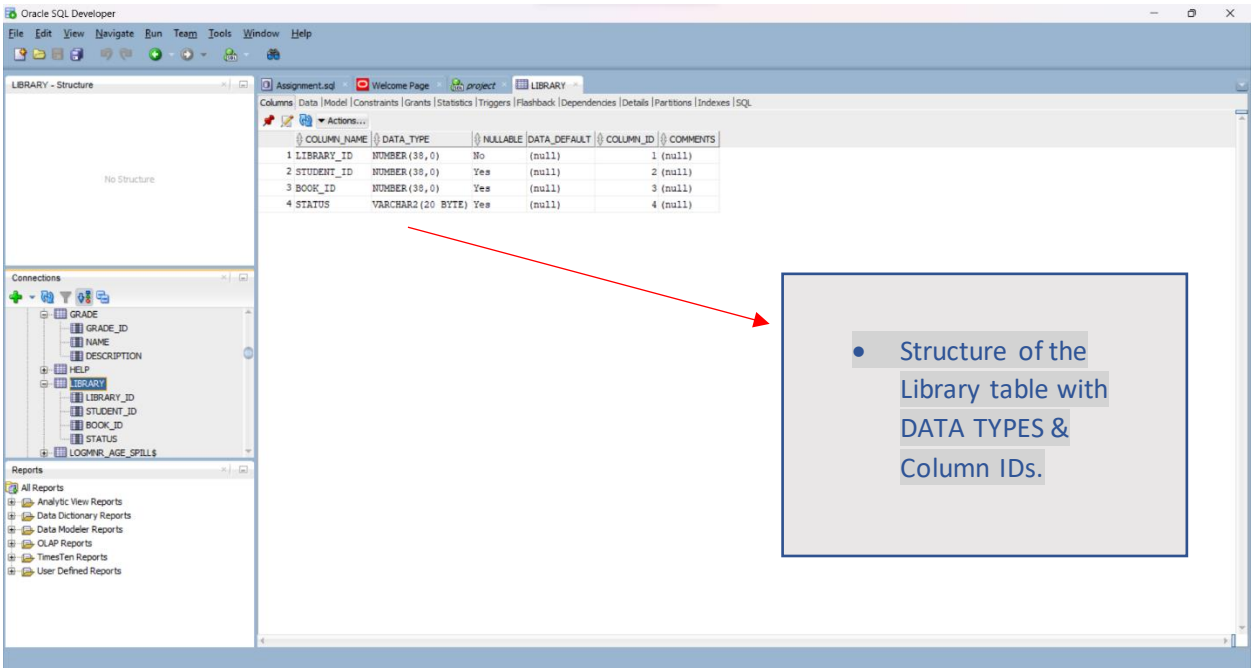
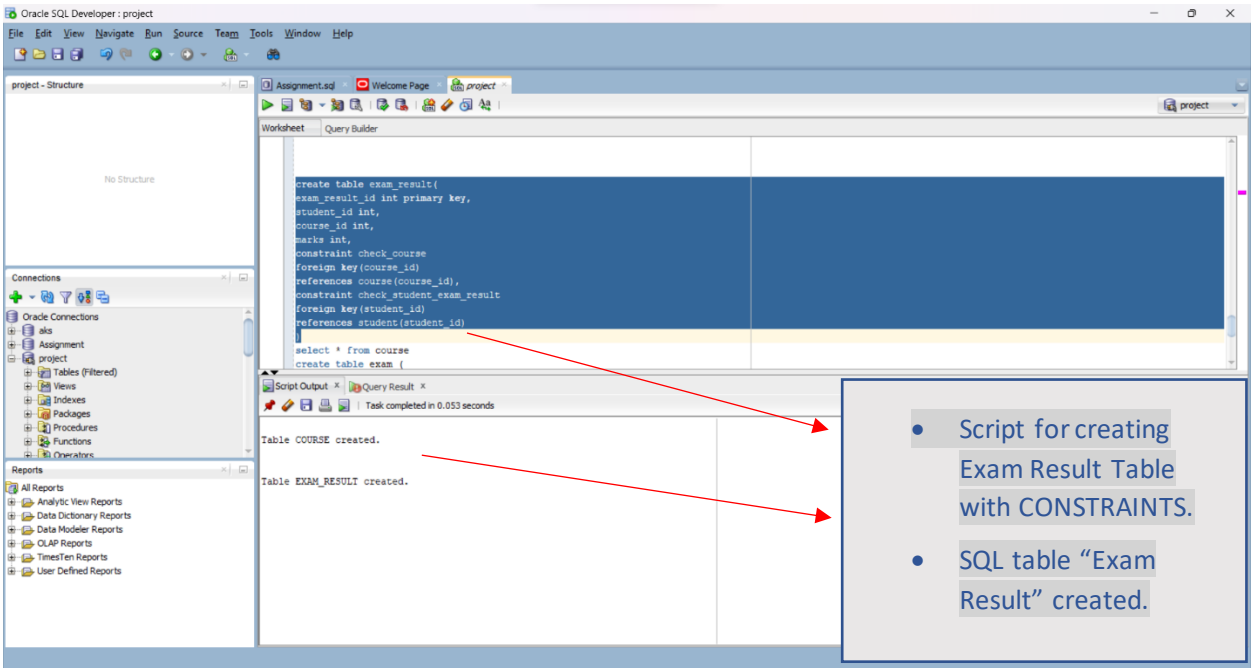


Table created for exam result:



Structure for exam result:

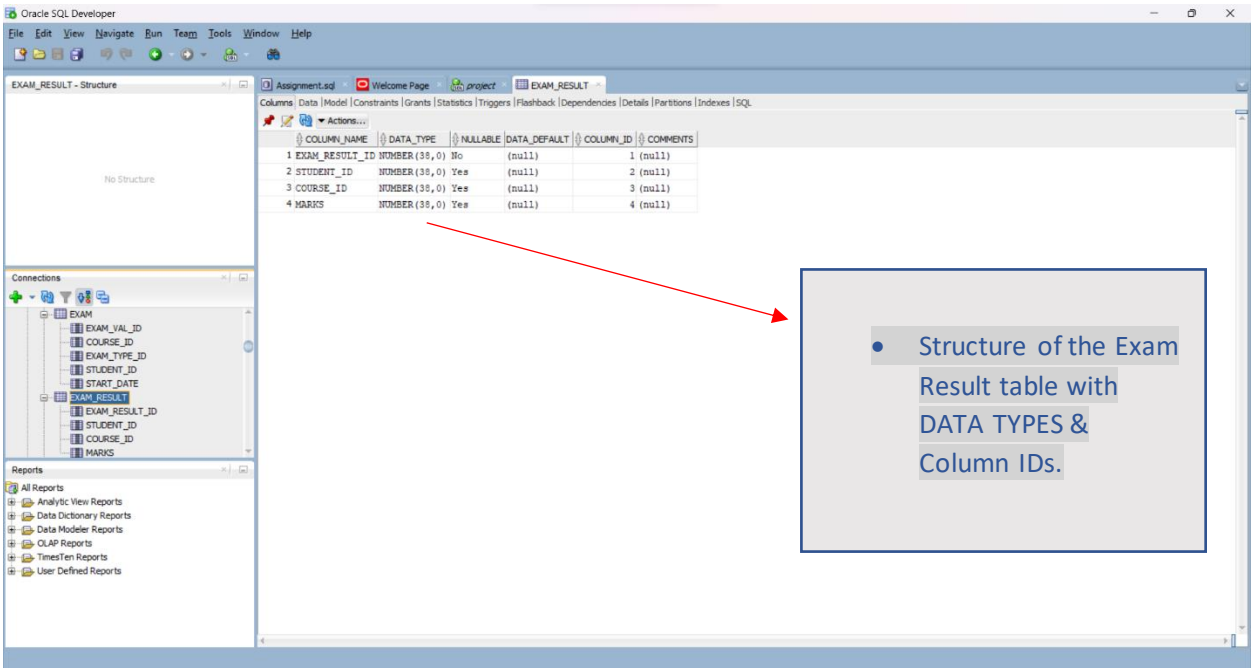
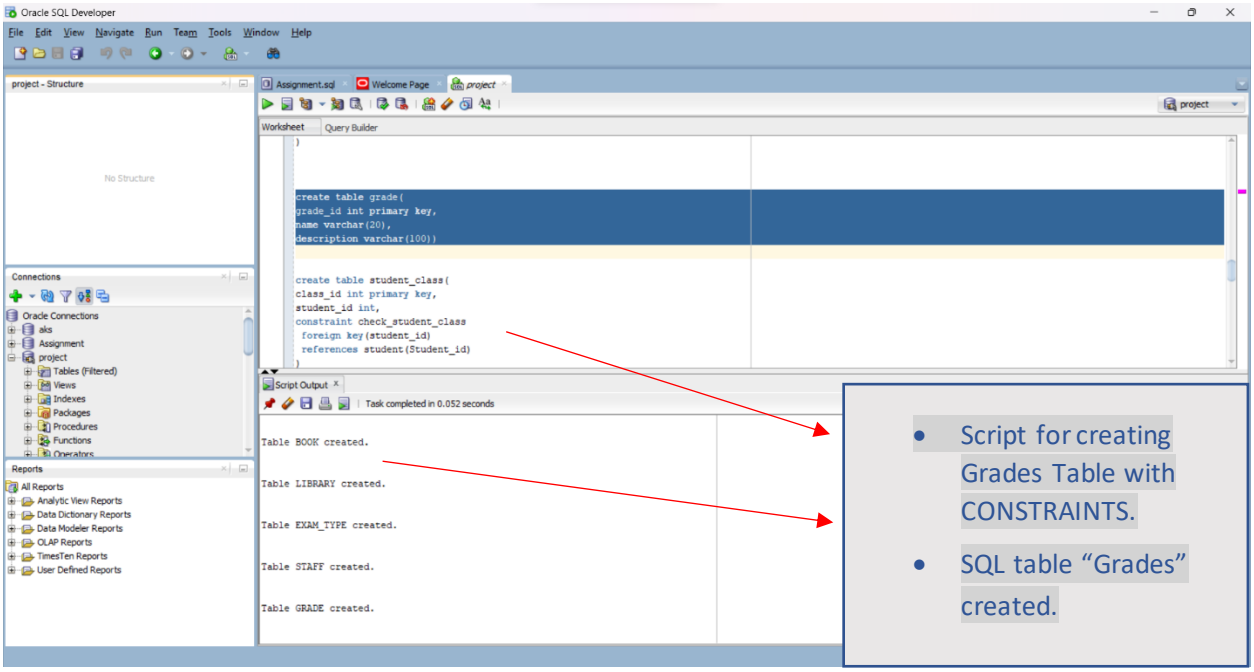


Table created for grades:



Structure for grade table:

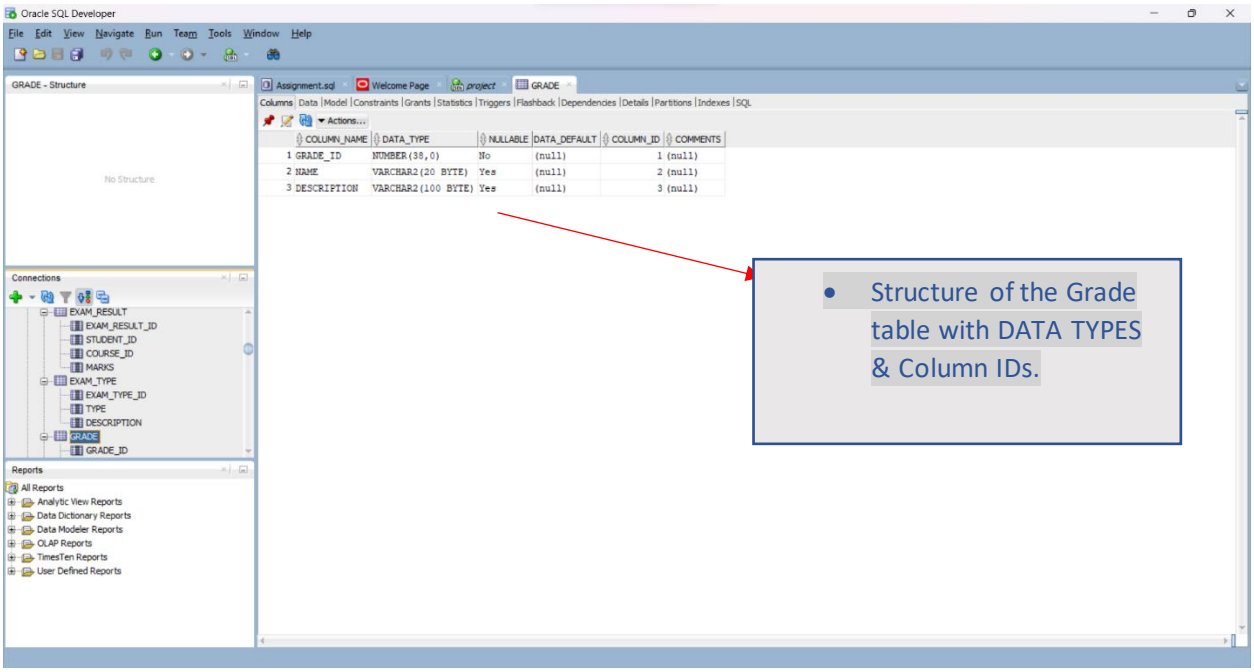
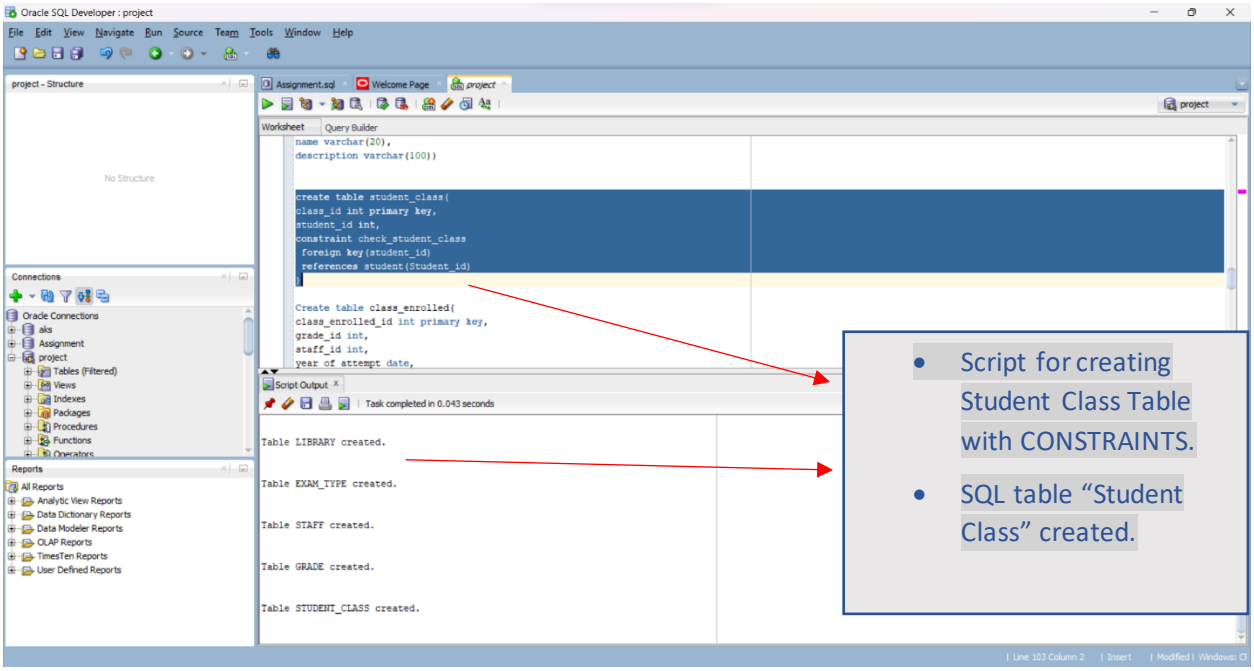
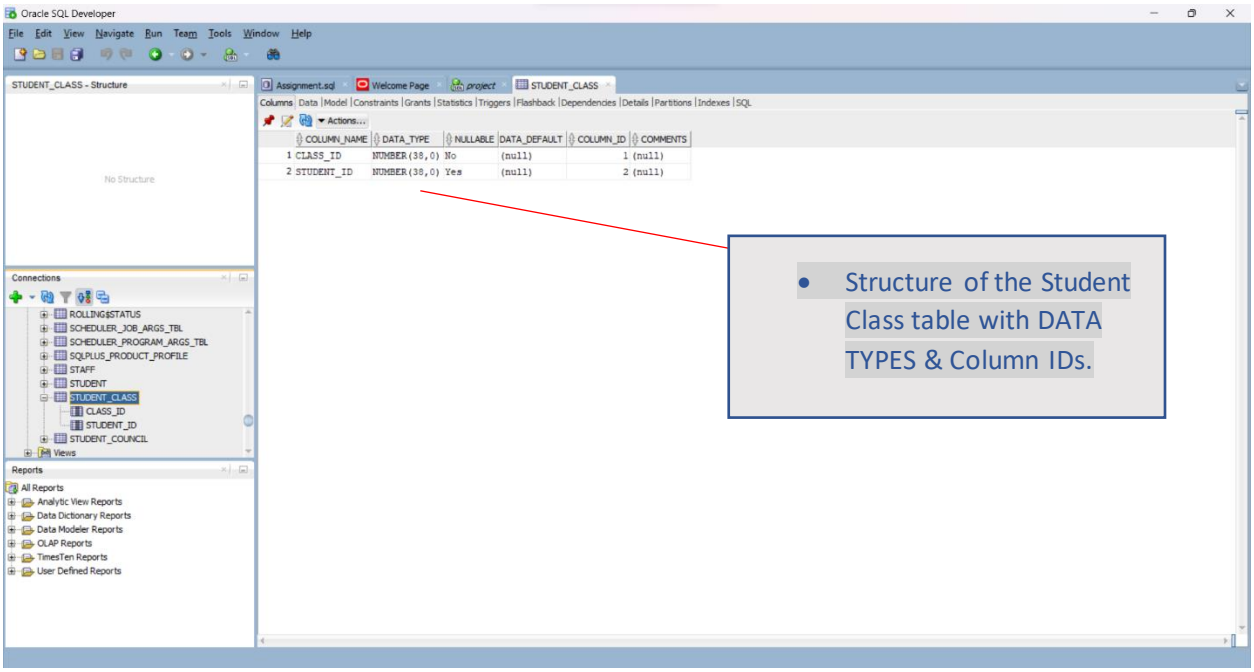


Table created for student class:



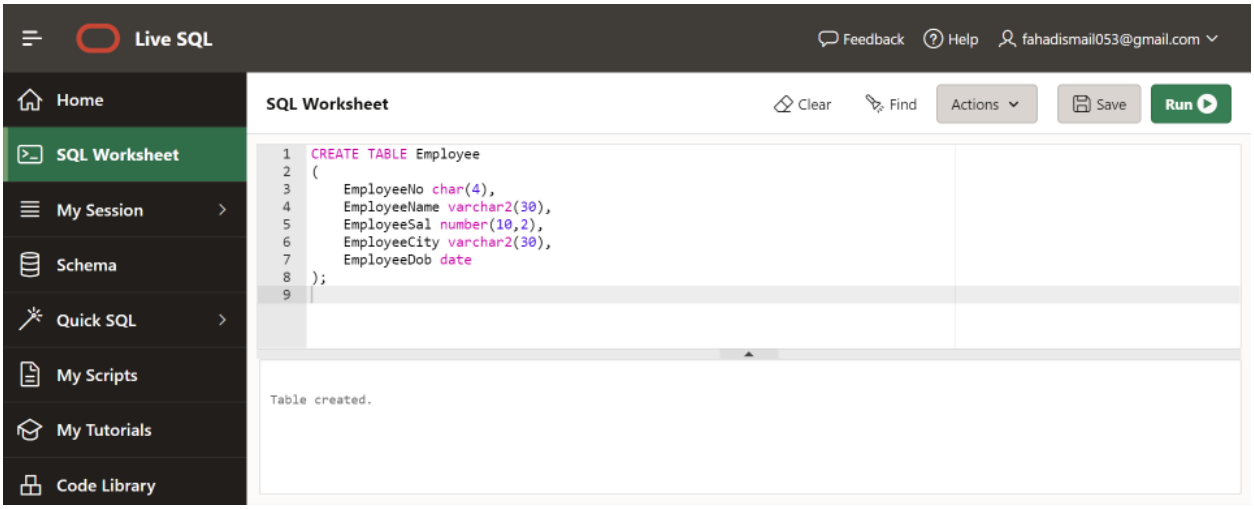
Structure for Student class:



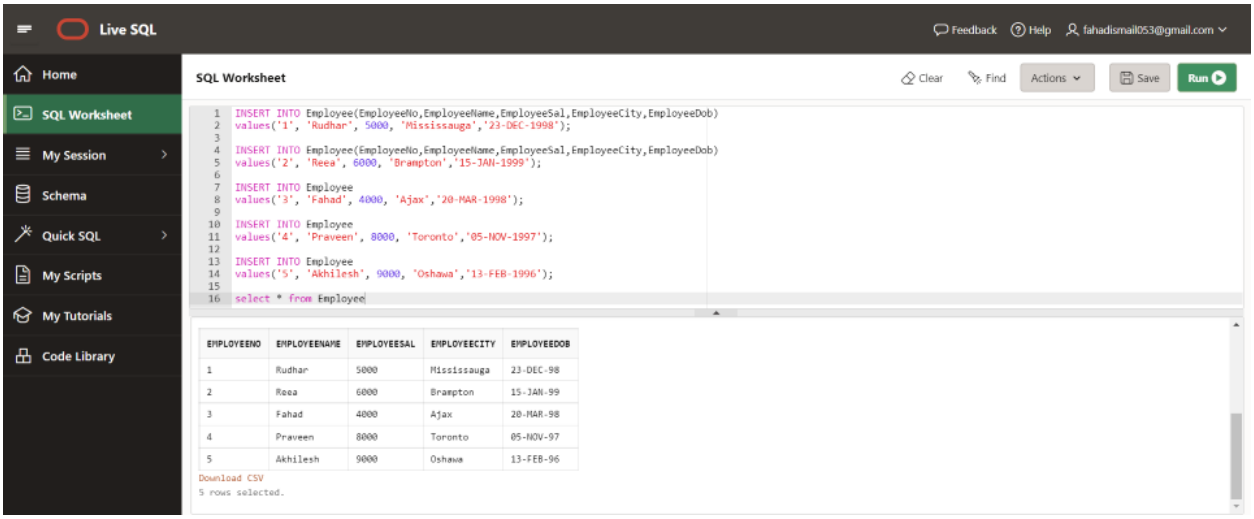
3. Physical Database Design

- Create a table of your choice and do the following operations
- CREATE (enter) 5 records to the table using SQL Script
- UPDATE one or more records (Using Script)
- DELETE one or more records (Using Script)
- DROP the table you created.
- Do not use your project tables for this activity.

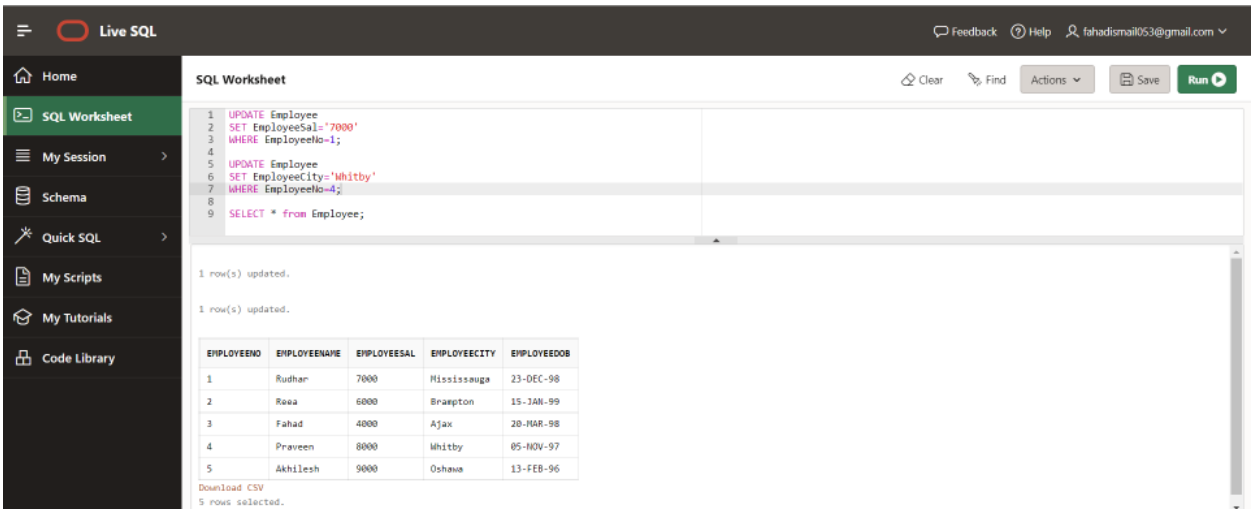
- Create Table



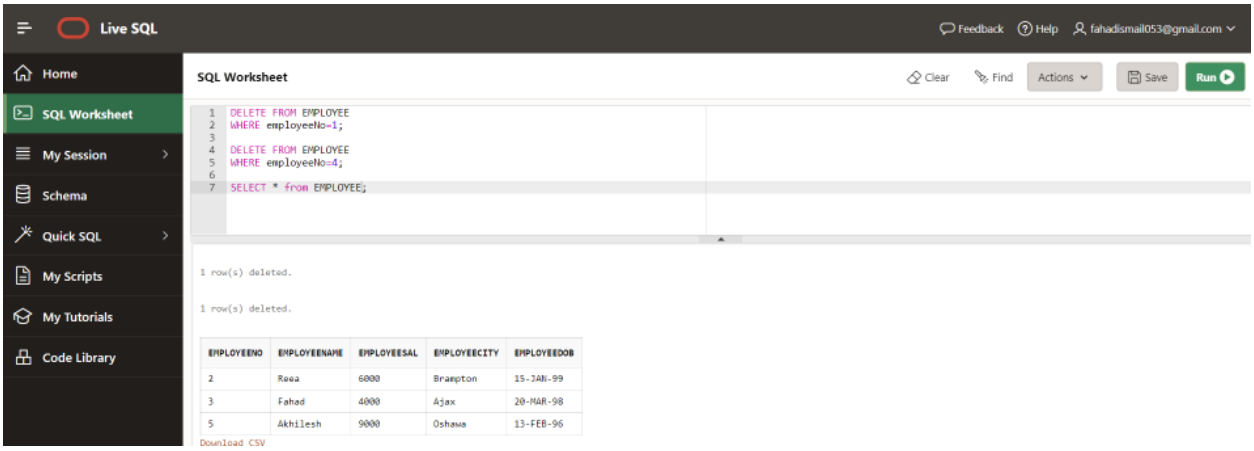
- Enter 5 records to the table



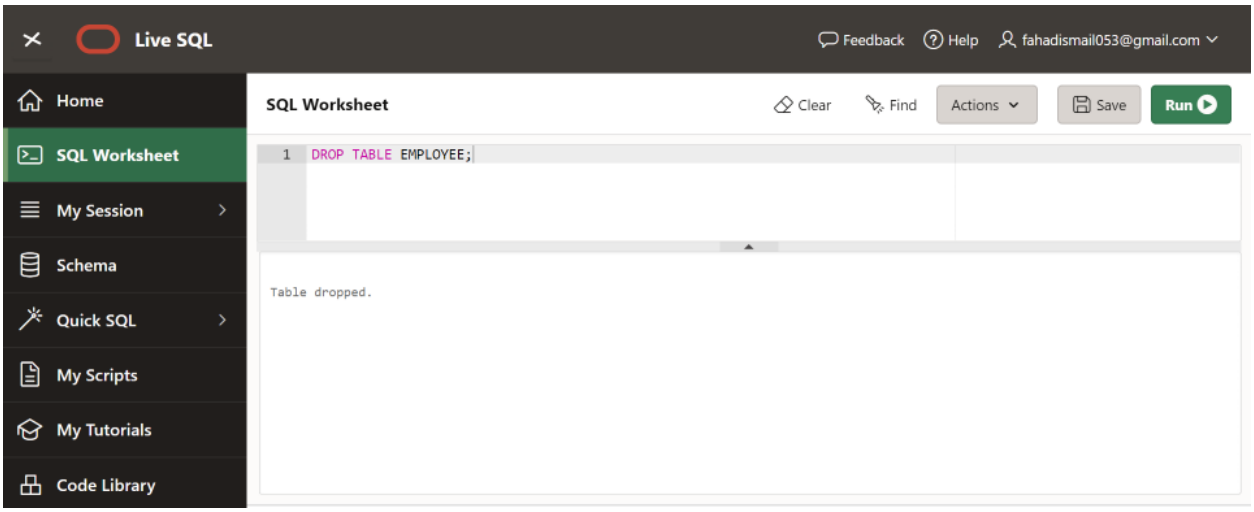
- Update One or More records:



- Delete One or More Records:



- Drop the Table:


Live SQL
Feedback
Help
fahadismail053@gmail.com
Home
SQL Worksheet
My Session
Schema
Quick SQL
My Scripts
My Tutorials
Code Library
SQL Worksheet
Clear
Find
Actions
Save
Run
1
DROP TABLE EMPLOYEE;
Table dropped.

CONCLUSION

Time to wrap it up. What is your conclusion? How would you synthesize all the information into something even the busiest CEO wants to read? What are the key takeaways? How does your product/service/methodology uniquely address the issues raised by your study?

Key Takeaways

1. ***Better Communication***

A student database management system allows for easier communication between students, parents-teachers, students-teachers and even allows for the school alumni to stay connected with each other.

Instead of only relying on the school diary, the teacher can send instant updates to the parents related to school events, attendance, and even disciplinary issues over email or SMS. Students can connect with each other and with their teachers over on the discussion boards to clear their doubts and collaborate over assignments.

2. ***Simplified Admission Process***

The admission season can be a chaotic time of the year for every school, but with the aid of the student database management system, the process gets simplified for everyone involved. From managing applicant database online to accepting registrations etc. the prospective students can use the cloud-based system to submit the forms, upload essential documents, apply for scholarships, etc.

The online system also makes it easy for the school to send students updates on admission tests, interview dates, reminders as well as the final admission status. This makes the admission process faster, error-free and more secure.

3. ***Faster Attendance Management***

The attendance management system uses a biometric system or an access card to maintain a quick and accurate record of the student and school staff attendance. The system saves class teachers tedious minutes every morning calling out the class attendance and provides with the administration team on timely summaries and records of regular latecomers, absentees etc. when needed.

The system also allows for attendance updates to be sent to parents. It enables schools to track the attendance of the teachers and the various staff and calculate leaves and working days with ease and accuracy.