23M --Advanced Database Concepts (SEC. 001)

Professor: Arpit Singh

Date: Aug 4, 2023

Group 9

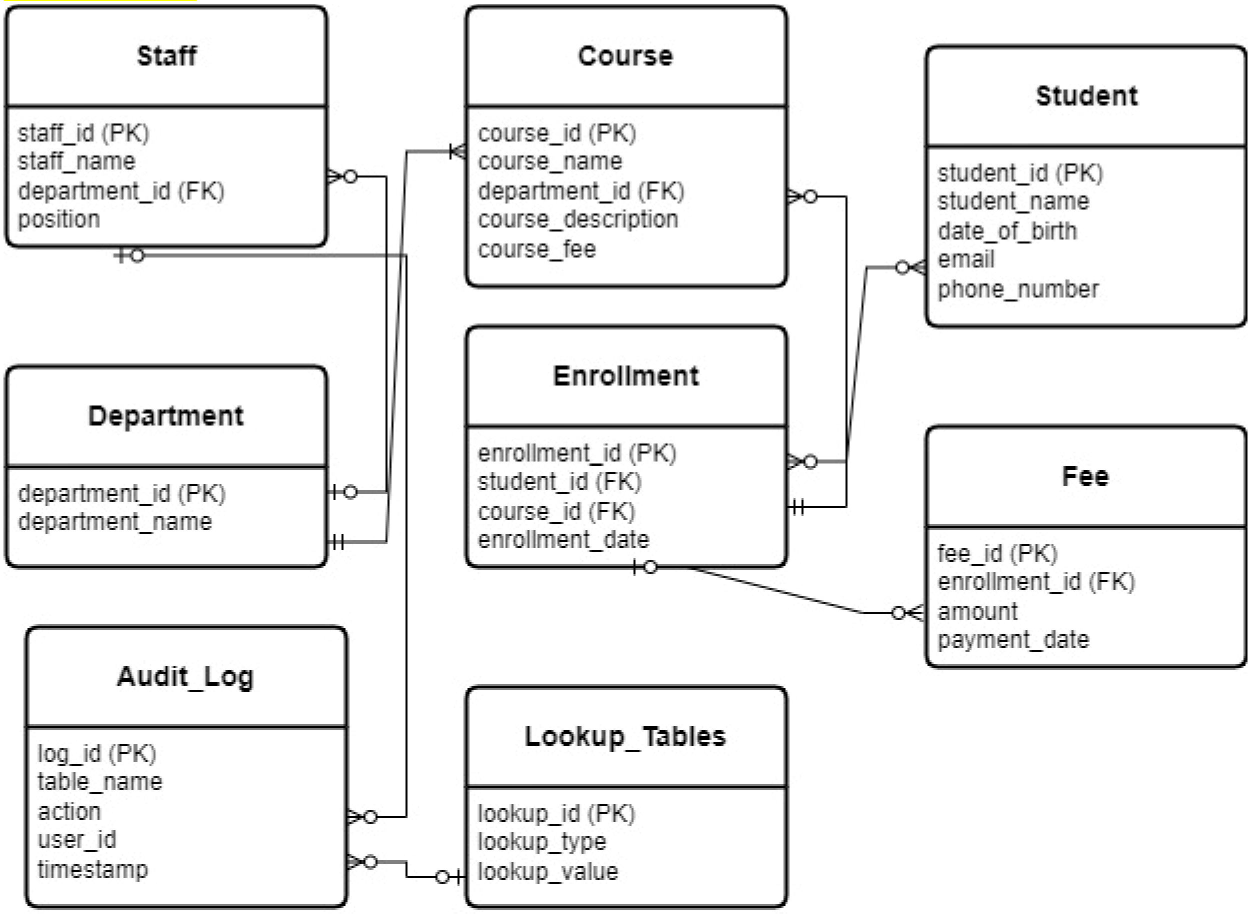
Student Name: LING LIN	**Student ID: 301292283**
Student Name: Yu-Wen, Hung	**Student ID:301253941**

1. Content of my database

```
-- Create Department table
CREATE TABLE Department (
   department_id NUMBER PRIMARY KEY,
   department_name VARCHAR2(255)
);

-- Create Course table
CREATE TABLE Course (
   course_id NUMBER PRIMARY KEY,
   course_name VARCHAR2(255),
   department_id NUMBER,
   course_description CLOB,
   course_fee NUMBER(10, 2),
   CONSTRAINT fk_course_department FOREIGN KEY (department_id)
REFERENCES Department (department_id)
);

-- Create a Student table
CREATE TABLE Student (
   student_id NUMBER PRIMARY KEY,
   student_name VARCHAR2(255),
   date_of_birth DATE,
   email VARCHAR2(255),
   phone_number VARCHAR2(20)
);

-- Create Enrollment form
CREATE TABLE Enrollment (
   enrollment_id NUMBER PRIMARY KEY,
   student_id NUMBER,
   course_id NUMBER,
   enrollment_date DATE,
    CONSTRAINT fk_enrollment_student FOREIGN KEY (student_id)
REFERENCES Student (student_id),
    CONSTRAINT fk_enrollment_course FOREIGN KEY (course_id)
REFERENCES Course (course_id)
);

-- Create Fee form
CREATE TABLE Fee (
    fee_id NUMBER PRIMARY KEY,
    enrollment_id NUMBER,
    amount NUMBER(10, 2),
    payment_date DATE,
    CONSTRAINT fk_fee_enrollment FOREIGN KEY (enrollment_id)
REFERENCES Enrollment (enrollment_id)
);

-- Create Staff table
CREATE TABLE Staff (
    staff_id NUMBER PRIMARY KEY,
    staff_name VARCHAR2(255),
    department_id NUMBER,
    position VARCHAR2(100),
    CONSTRAINT fk_staff_department FOREIGN KEY (department_id)
REFERENCES Department (department_id)
);

-- Create the Audit Log table
CREATE TABLE Audit_Log (
    log_id NUMBER PRIMARY KEY,
    table_name VARCHAR2(255),
    action VARCHAR2(10),
    user_id NUMBER,
    timestamp TIMESTAMP
);
```

```
-- Create the Lookup Tables table
CREATE TABLE Lookup_Tables (
   lookup_id NUMBER PRIMARY KEY,
   lookup_type VARCHAR2(50),
   lookup_value VARCHAR2(255)
);
```

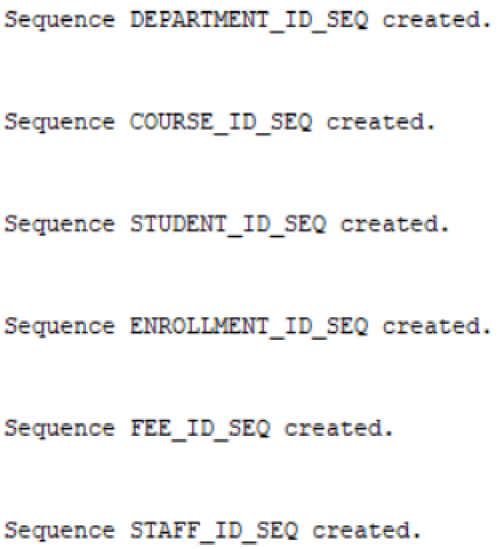
--ER diagram



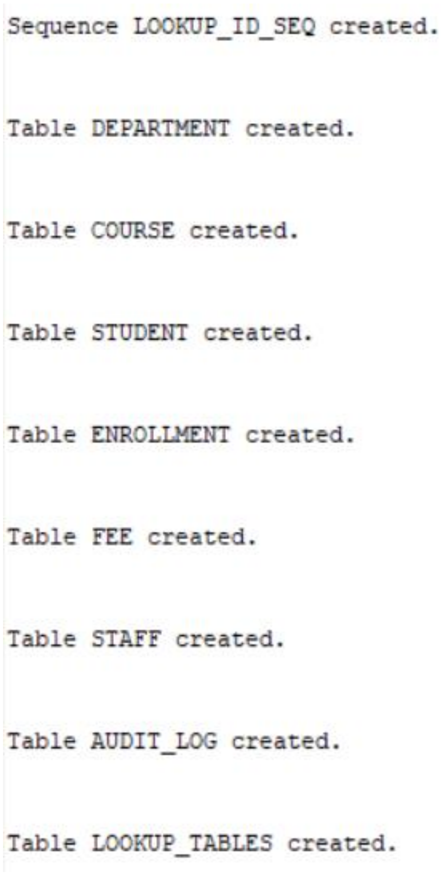
2. Sequence

```
-- Create at least one sequence
CREATE SEQUENCE department_id_seq START WITH 1 INCREMENT BY 1;
CREATE SEQUENCE course_id_seq START WITH 1 INCREMENT BY 1;
CREATE SEQUENCE student_id_seq START WITH 1 INCREMENT BY 1;
CREATE SEQUENCE enrollment_id_seq START WITH 1 INCREMENT BY 1;
CREATE SEQUENCE fee_id_seq START WITH 1 INCREMENT BY 1;
```

CREATE SEQUENCE staff_id_seq START WITH 1 INCREMENT BY 1;
CREATE SEQUENCE audit_log_id_seq START WITH 1 INCREMENT BY 1;
CREATE SEQUENCE lookup_id_seq START WITH 1 INCREMENT BY 1;



```
Sequence DEPARTMENT_ID_SEQ created.


Sequence COURSE_ID_SEQ created.


Sequence STUDENT_ID_SEQ created.


Sequence ENROLLMENT_ID_SEQ created.


Sequence FEE_ID_SEQ created.


Sequence STAFF_ID_SEQ created.
```



```
Sequence LOOKUP_ID_SEQ created.


Table DEPARTMENT created.


Table COURSE created.


Table STUDENT created.


Table ENROLLMENT created.


Table FEE created.


Table STAFF created.


Table AUDIT_LOG created.


Table LOOKUP_TABLES created.
```

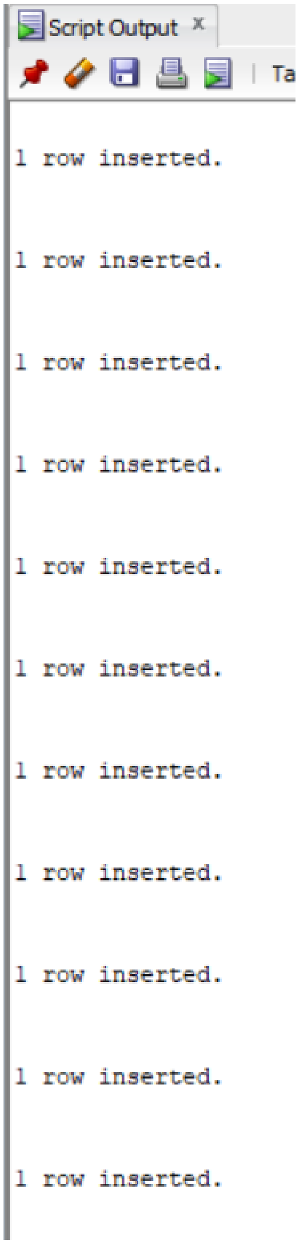
```
-- Inserting data into Department table
INSERT INTO Department (department_id, department_name) VALUES
(department_id_seq.NEXTVAL, 'Computer Science');
INSERT INTO Department (department_id, department_name) VALUES
(department_id_seq.NEXTVAL, 'Business');
INSERT INTO Department (department_id, department_name) VALUES
(department_id_seq.NEXTVAL, 'Engineering');
INSERT INTO Department (department_id, department_name) VALUES
(department_id_seq.NEXTVAL, 'Mathematics');
INSERT INTO Department (department_id, department_name) VALUES
(department_id_seq.NEXTVAL, 'Physics');
INSERT INTO Department (department_id, department_name) VALUES
(department_id_seq.NEXTVAL, 'History');
INSERT INTO Department (department_id, department_name) VALUES
(department_id_seq.NEXTVAL, 'Chemistry');
INSERT INTO Department (department_id, department_name) VALUES
(department_id_seq.NEXTVAL, 'Literature');
INSERT INTO Department (department_id, department_name) VALUES
(department_id_seq.NEXTVAL, 'Economics');
INSERT INTO Department (department_id, department_name) VALUES
(department_id_seq.NEXTVAL, 'Psychology');


-- Inserting data into Course table
INSERT INTO Course (course_id, course_name, department_id, course_description,
course_fee) VALUES
   (course_id_seq.NEXTVAL, 'Introduction to Programming', 1, 'Learn the basics of
programming.', 1000.00);
INSERT INTO Course (course_id, course_name, department_id, course_description,
course_fee) VALUES
   (course_id_seq.NEXTVAL, 'Database Management', 1, 'Introduction to database
systems.', 1200.00);
INSERT INTO Course (course_id, course_name, department_id, course_description,
course_fee) VALUES
   (course_id_seq.NEXTVAL, 'Marketing Fundamentals', 2, 'Principles of marketing.',
950.00);
INSERT INTO Course (course_id, course_name, department_id, course_description,
course_fee) VALUES
   (course_id_seq.NEXTVAL, 'Advanced Mathematics', 3, 'Advanced concepts in
mathematics.', 1300.00);
```

```
INSERT INTO Course (course_id, course_name, department_id, course_description,
course_fee) VALUES
   (course_id_seq.NEXTVAL, 'Quantum Physics', 4, 'Exploring the principles of quantum
physics.', 1100.00);
INSERT INTO Course (course_id, course_name, department_id, course_description,
course_fee) VALUES
   (course_id_seq.NEXTVAL, 'World History', 5, 'A journey through world history.',
900.00);
INSERT INTO Course (course_id, course_name, department_id, course_description,
course_fee) VALUES
   (course_id_seq.NEXTVAL, 'Organic Chemistry', 6, 'Understanding organic compounds.',
1250.00);
INSERT INTO Course (course_id, course_name, department_id, course_description,
course_fee) VALUES
   (course_id_seq.NEXTVAL, 'Literary Analysis', 7, 'Analyzing literature from different
eras.', 850.00);
INSERT INTO Course (course_id, course_name, department_id, course_description,
course_fee) VALUES
   (course_id_seq.NEXTVAL, 'Macroeconomics', 8, 'Studying national economies.',
1050.00);
INSERT INTO Course (course_id, course_name, department_id, course_description,
course_fee) VALUES
   (course_id_seq.NEXTVAL, 'Psychological Research', 9, 'Exploring methods in
psychological research.', 1150.00);


-- Inserting data into Student table
INSERT INTO Student (student_id, student_name, date_of_birth, email, phone_number)
VALUES
   (student_id_seq.NEXTVAL, 'John Doe', TO_DATE('1995-01-15', 'YYYY-MM-DD'),
'john@example.com', '123-456-7890');
INSERT INTO Student (student_id, student_name, date_of_birth, email, phone_number)
VALUES
   (student_id_seq.NEXTVAL, 'Jane Smith', TO_DATE('1998-05-20', 'YYYY-MM-DD'),
'jane@example.com', '987-654-3210');
INSERT INTO Student (student_id, student_name, date_of_birth, email, phone_number)
VALUES
   (student_id_seq.NEXTVAL, 'Michael Johnson', TO_DATE('1993-09-10', 'YYYY-MM-DD'),
'michael@example.com', '456-789-0123');
INSERT INTO Student (student_id, student_name, date_of_birth, email, phone_number)
VALUES
```

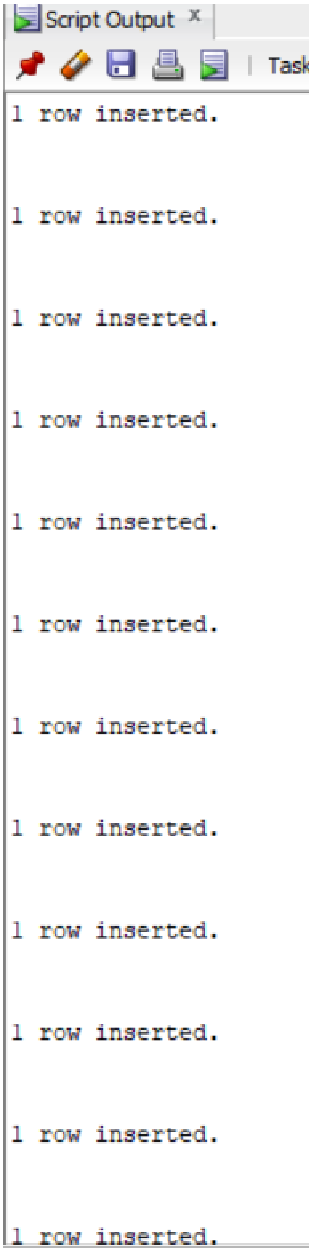
```
   (student_id_seq.NEXTVAL, 'Emily Williams', TO_DATE('1997-07-03', 'YYYY-MM-DD'),
'emily@example.com', '789-012-3456');
INSERT INTO Student (student_id, student_name, date_of_birth, email, phone_number)
VALUES
   (student_id_seq.NEXTVAL, 'David Brown', TO_DATE('1994-03-25', 'YYYY-MM-DD'),
'david@example.com', '234-567-8901');
INSERT INTO Student (student_id, student_name, date_of_birth, email, phone_number)
VALUES
   (student_id_seq.NEXTVAL, 'Olivia Miller', TO_DATE('1999-11-12', 'YYYY-MM-DD'),
'olivia@example.com', '567-890-1234');
INSERT INTO Student (student_id, student_name, date_of_birth, email, phone_number)
VALUES
   (student_id_seq.NEXTVAL, 'Sophia Wilson', TO_DATE('1996-02-28', 'YYYY-MM-DD'),
'sophia@example.com', '890-123-4567');
INSERT INTO Student (student_id, student_name, date_of_birth, email, phone_number)
VALUES
   (student_id_seq.NEXTVAL, 'James Taylor', TO_DATE('1992-08-08', 'YYYY-MM-DD'),
'james@example.com', '123-456-7890');
INSERT INTO Student (student_id, student_name, date_of_birth, email, phone_number)
VALUES
   (student_id_seq.NEXTVAL, 'Emma Davis', TO_DATE('1997-04-17', 'YYYY-MM-DD'),
'emma@example.com', '456-789-0123');
INSERT INTO Student (student_id, student_name, date_of_birth, email, phone_number)
VALUES
   (student_id_seq.NEXTVAL, 'Liam Anderson', TO_DATE('1995-12-02', 'YYYY-MM-DD'),
'liam@example.com', '789-012-3456');


-- Inserting data into Enrollment table
INSERT INTO Enrollment (enrollment_id, student_id, course_id, enrollment_date)
VALUES
   (enrollment_id_seq.NEXTVAL, 2, 1, TO_DATE('2023-07-15', 'YYYY-MM-DD'));
INSERT INTO Enrollment (enrollment_id, student_id, course_id, enrollment_date)
VALUES
   (enrollment_id_seq.NEXTVAL, 3, 2, TO_DATE('2023-07-20', 'YYYY-MM-DD'));
INSERT INTO Enrollment (enrollment_id, student_id, course_id, enrollment_date)
VALUES
   (enrollment_id_seq.NEXTVAL, 4, 3, TO_DATE('2023-08-03', 'YYYY-MM-DD'));
INSERT INTO Enrollment (enrollment_id, student_id, course_id, enrollment_date)
VALUES
   (enrollment_id_seq.NEXTVAL, 5, 1, TO_DATE('2023-07-22', 'YYYY-MM-DD'));
```

```
INSERT INTO Enrollment (enrollment_id, student_id, course_id, enrollment_date)
VALUES
    (enrollment_id_seq.NEXTVAL, 6, 2, TO_DATE('2023-08-10', 'YYYY-MM-DD'));
INSERT INTO Enrollment (enrollment_id, student_id, course_id, enrollment_date)
VALUES
    (enrollment_id_seq.NEXTVAL, 7, 3, TO_DATE('2023-08-05', 'YYYY-MM-DD'));
INSERT INTO Enrollment (enrollment_id, student_id, course_id, enrollment_date)
VALUES
    (enrollment_id_seq.NEXTVAL, 8, 1, TO_DATE('2023-07-18', 'YYYY-MM-DD'));
INSERT INTO Enrollment (enrollment_id, student_id, course_id, enrollment_date)
VALUES
    (enrollment_id_seq.NEXTVAL, 9, 2, TO_DATE('2023-07-25', 'YYYY-MM-DD'));
INSERT INTO Enrollment (enrollment_id, student_id, course_id, enrollment_date)
VALUES
    (enrollment_id_seq.NEXTVAL, 10, 3, TO_DATE('2023-08-08', 'YYYY-MM-DD'));


-- Inserting data into Fee table
INSERT INTO Fee (fee_id, enrollment_id, amount, payment_date) VALUES
    (fee_id_seq.NEXTVAL, 4, 850.00, TO_DATE('2023-07-25', 'YYYY-MM-DD'));
INSERT INTO Fee (fee_id, enrollment_id, amount, payment_date) VALUES
    (fee_id_seq.NEXTVAL, 5, 1100.00, TO_DATE('2023-08-10', 'YYYY-MM-DD'));
INSERT INTO Fee (fee_id, enrollment_id, amount, payment_date) VALUES
    (fee_id_seq.NEXTVAL, 6, 950.00, TO_DATE('2023-08-05', 'YYYY-MM-DD'));
INSERT INTO Fee (fee_id, enrollment_id, amount, payment_date) VALUES
    (fee_id_seq.NEXTVAL, 7, 1200.00, TO_DATE('2023-07-18', 'YYYY-MM-DD'));
INSERT INTO Fee (fee_id, enrollment_id, amount, payment_date) VALUES
    (fee_id_seq.NEXTVAL, 8, 1050.00, TO_DATE('2023-07-25', 'YYYY-MM-DD'));
INSERT INTO Fee (fee_id, enrollment_id, amount, payment_date) VALUES
    (fee_id_seq.NEXTVAL, 9, 800.00, TO_DATE('2023-08-08', 'YYYY-MM-DD'));
INSERT INTO Fee (fee_id, enrollment_id, amount, payment_date) VALUES
    (fee_id_seq.NEXTVAL, 10, 1150.00, TO_DATE('2023-08-01', 'YYYY-MM-DD'));


-- Inserting data into Staff table
INSERT INTO Staff (staff_id, staff_name, department_id, position) VALUES
    (staff_id_seq.NEXTVAL, 'Sarah Anderson', 1, 'Assistant Professor');
INSERT INTO Staff (staff_id, staff_name, department_id, position) VALUES
    (staff_id_seq.NEXTVAL, 'David Smith', 2, 'Lecturer');
INSERT INTO Staff (staff_id, staff_name, department_id, position) VALUES
    (staff_id_seq.NEXTVAL, 'Jennifer Lee', 3, 'Associate Professor');
```

```
INSERT INTO Staff (staff_id, staff_name, department_id, position) VALUES
    (staff_id_seq.NEXTVAL, 'Robert Johnson', 1, 'Professor');
INSERT INTO Staff (staff_id, staff_name, department_id, position) VALUES
    (staff_id_seq.NEXTVAL, 'Karen Brown', 2, 'Lecturer');



-- Inserting data into Lookup_Tables table
INSERT INTO Lookup_Tables (lookup_id, lookup_type, lookup_value) VALUES
    (lookup_id_seq.NEXTVAL, 'Department Type', 'Science');
INSERT INTO Lookup_Tables (lookup_id, lookup_type, lookup_value) VALUES
    (lookup_id_seq.NEXTVAL, 'Department Type', 'Business');
INSERT INTO Lookup_Tables (lookup_id, lookup_type, lookup_value) VALUES
    (lookup_id_seq.NEXTVAL, 'Department Type', 'Engineering');
INSERT INTO Lookup_Tables (lookup_id, lookup_type, lookup_value) VALUES
    (lookup_id_seq.NEXTVAL, 'Department Type', 'Arts');
INSERT INTO Lookup_Tables (lookup_id, lookup_type, lookup_value) VALUES
    (lookup_id_seq.NEXTVAL, 'Course Level', 'Beginner');
```

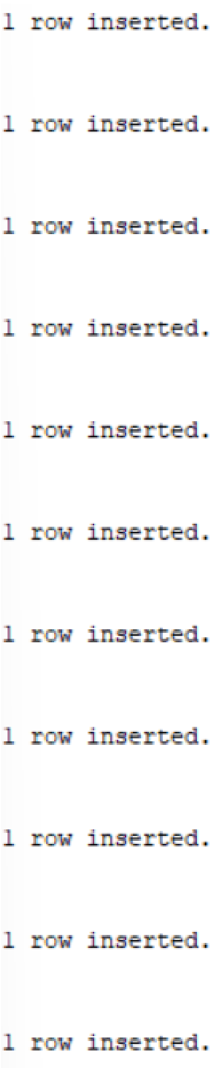


```
1 row inserted.


1 row inserted.


1 row inserted.


1 row inserted.


1 row inserted.


1 row inserted.


1 row inserted.


1 row inserted.


1 row inserted.


1 row inserted.


1 row inserted.
```



```
1 row inserted.


1 row inserted.


1 row inserted.


1 row inserted.


1 row inserted.


1 row inserted.


1 row inserted.


1 row inserted.


1 row inserted.


1 row inserted.


1 row inserted.


1 row inserted.
```

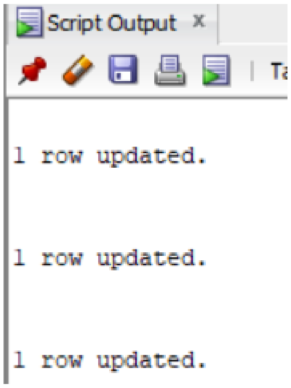
```
1 row inserted.


1 row inserted.


1 row inserted.


1 row inserted.


1 row inserted.


1 row inserted.


1 row inserted.


1 row inserted.


1 row inserted.


1 row inserted.
```



```
1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.
```

```
-- Update table(s)
-- Update the enrollment_date field in the Enrollment table
UPDATE Enrollment
SET enrollment_date = TO_DATE('2023-09-01', 'YYYY-MM-DD')
WHERE enrollment_id = 1;

-- Update the course_fee field in the Course table
UPDATE Course
SET course_fee = 1500.00
WHERE course_id = 1;

-- Update the email field in the Student table
UPDATE Student
SET email = 'updated_email@example.com'
```

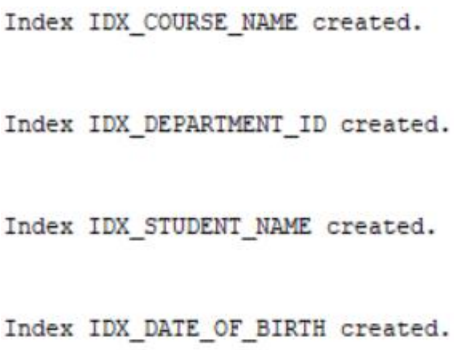
```
WHERE student_id = 1;
```



```
1 row updated.


1 row updated.


1 row updated.
```

--3. Indexes

```
-- Index for Course table
CREATE INDEX idx_course_name ON Course(course_name);
CREATE INDEX idx_department_id ON Course(department_id);

-- Index for Student table
CREATE INDEX idx_student_name ON Student(student_name);
CREATE INDEX idx_date_of_birth ON Student(date_of_birth);
```



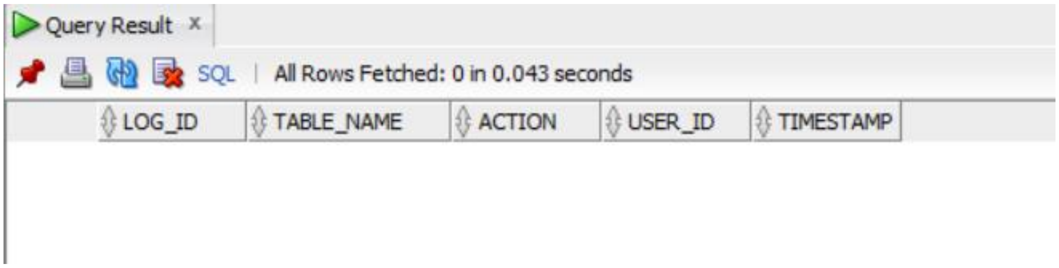
```
Index IDX_COURSE_NAME created.


Index IDX_DEPARTMENT_ID created.


Index IDX_STUDENT_NAME created.


Index IDX_DATE_OF_BIRTH created.
```

-- 4. Trigger

-- Trigger 1: Audit Log Trigger

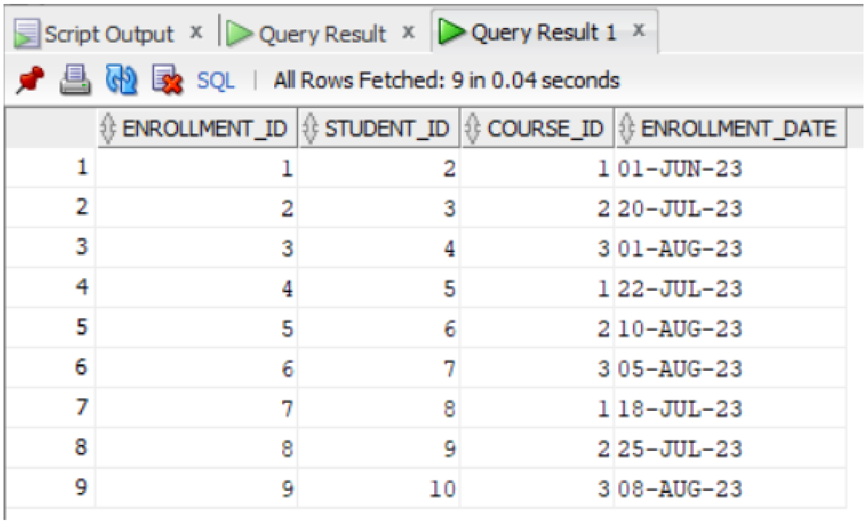
```
CREATE OR REPLACE TRIGGER trg_audit_log
AFTER INSERT OR UPDATE OR DELETE ON Enrollment
FOR EACH ROW
BEGIN
   IF INSERTING THEN
      INSERT INTO Audit_Log(log_id, table_name, action, user_id, timestamp)
```

```
    VALUES(audit_log_id_seq.NEXTVAL, 'Enrollment', 'INSERT', USER, SYSTIMESTAMP);
  ELSIF UPDATING THEN
    INSERT INTO Audit_Log(log_id, table_name, action, user_id, timestamp)
    VALUES(audit_log_id_seq.NEXTVAL, 'Enrollment', 'UPDATE', USER, SYSTIMESTAMP);
  ELSIF DELETING THEN
    INSERT INTO Audit_Log(log_id, table_name, action, user_id, timestamp)
    VALUES(audit_log_id_seq.NEXTVAL, 'Enrollment', 'DELETE', USER, SYSTIMESTAMP);
  END IF;
  COMMIT;
END;
/
```



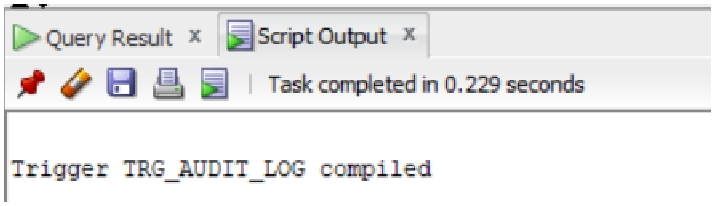
All Rows Fetched: 0 in 0.043 seconds

LOG_ID	TABLE_NAME	ACTION	USER_ID	TIMESTAMP



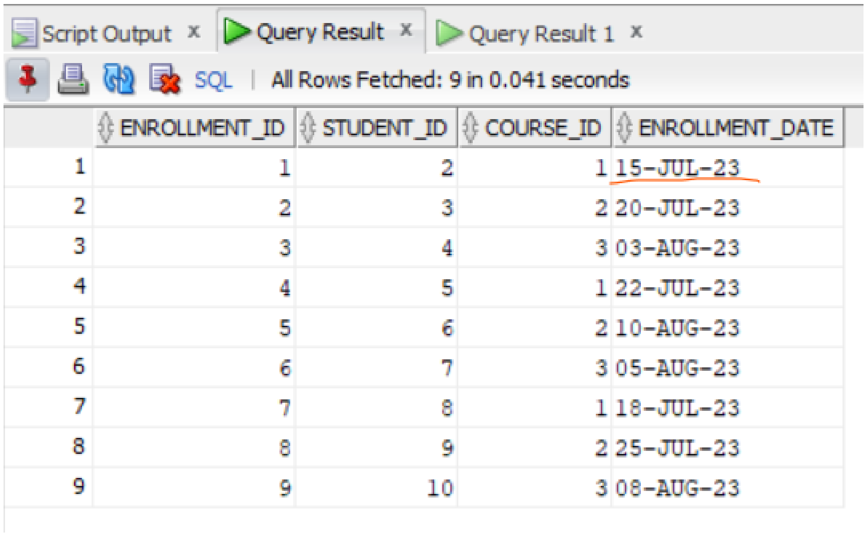
All Rows Fetched: 9 in 0.04 seconds

	ENROLLMENT_ID	STUDENT_ID	COURSE_ID	ENROLLMENT_DATE
1	1	2	1	01-JUN-23
2	2	3	2	20-JUL-23
3	3	4	3	01-AUG-23
4	4	5	1	22-JUL-23
5	5	6	2	10-AUG-23
6	6	7	3	05-AUG-23
7	7	8	1	18-JUL-23
8	8	9	2	25-JUL-23
9	9	10	3	08-AUG-23



Task completed in 0.229 seconds

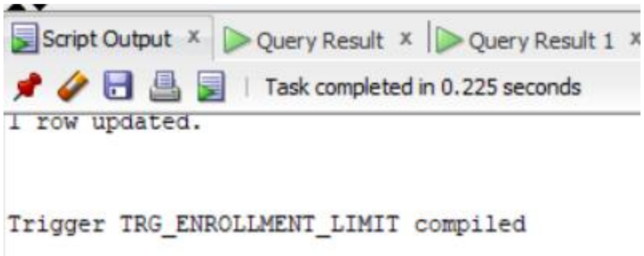
```
Trigger TRG_AUDIT_LOG compiled
```



All Rows Fetched: 9 in 0.041 seconds

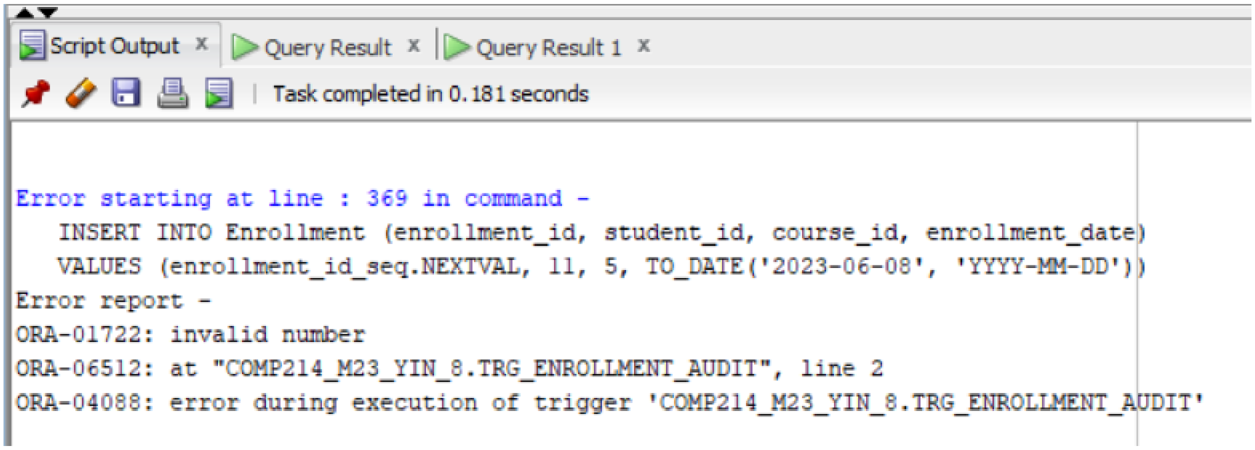
	ENROLLMENT_ID	STUDENT_ID	COURSE_ID	ENROLLMENT_DATE
1	1	2	1	15-JUL-23
2	2	3	2	20-JUL-23
3	3	4	3	03-AUG-23
4	4	5	1	22-JUL-23
5	5	6	2	10-AUG-23
6	6	7	3	05-AUG-23
7	7	8	1	18-JUL-23
8	8	9	2	25-JUL-23
9	9	10	3	08-AUG-23

```
-- Trigger 2: Enrollment Limit Trigger

CREATE OR REPLACE TRIGGER trg_enrollment_limit

BEFORE INSERT ON Enrollment

FOR EACH ROW

DECLARE

  current_enrollment NUMBER;

  max_capacity NUMBER;

BEGIN

  SELECT COUNT(*) INTO current_enrollment FROM Enrollment WHERE course_id = :NEW.course_id;

  SELECT course_fee INTO max_capacity FROM Course WHERE course_id = :NEW.course_id;


  IF current_enrollment >= max_capacity THEN

    RAISE_APPLICATION_ERROR(-20001, 'Course enrollment limit exceeded');

  END IF;

END;

/
```



```
1 row updated.


Trigger TRG_ENROLLMENT_LIMIT compiled
```

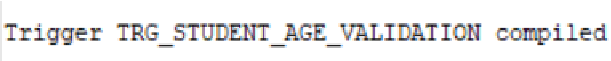
```
-- Insert a new record into the Enrollment table

  INSERT INTO Enrollment (enrollment_id, student_id, course_id, enrollment_date)

  VALUES (enrollment_id_seq.NEXTVAL, 11, 5, TO_DATE('2023-06-08', 'YYYY-MM-DD'));
```



```
Error starting at line : 369 in command -
   INSERT INTO Enrollment (enrollment_id, student_id, course_id, enrollment_date)
   VALUES (enrollment_id_seq.NEXTVAL, 11, 5, TO_DATE('2023-06-08', 'YYYY-MM-DD'))
Error report -
ORA-01722: invalid number
ORA-06512: at "COMP214_M23_YIN_8.TRG_ENROLLMENT_AUDIT", line 2
ORA-04088: error during execution of trigger 'COMP214_M23_YIN_8.TRG_ENROLLMENT_AUDIT'
```

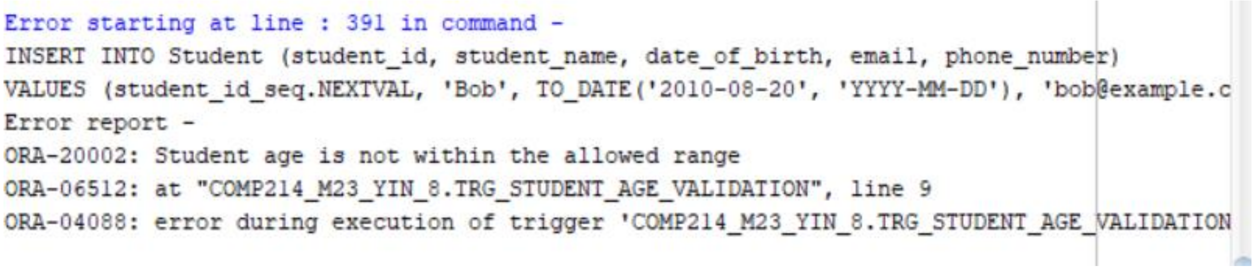
```
-- Trigger 3: Student Age Validation Trigger

CREATE OR REPLACE TRIGGER trg_student_age_validation

BEFORE INSERT OR UPDATE ON Student

FOR EACH ROW

DECLARE

   min_age NUMBER := 18; -- Minimum allowed age

   max_age NUMBER := 30; -- Maximum allowed age

   student_age NUMBER;

BEGIN

   student_age := TRUNC(MONTHS_BETWEEN(SYSDATE, :NEW.date_of_birth) / 12);


   IF student_age < min_age OR student_age > max_age THEN
```

```
        RAISE_APPLICATION_ERROR(-20002, 'Student age is not within the allowed range');
    END IF;
END;
/
```



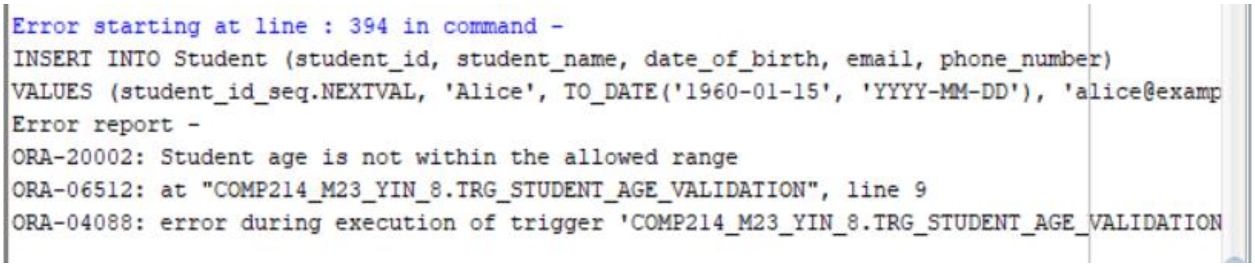
```
Trigger TRG_STUDENT_AGE_VALIDATION compiled
```

```
INSERT INTO Student (student_id, student_name, date_of_birth, email, phone_number)
VALUES (student_id_seq.NEXTVAL, 'Bob', TO_DATE('2010-08-20', 'YYYY-MM-DD'), 'bob@example.com', '987-654-3210');
```



```
Error starting at line : 391 in command -
INSERT INTO Student (student_id, student_name, date_of_birth, email, phone_number)
VALUES (student_id_seq.NEXTVAL, 'Bob', TO_DATE('2010-08-20', 'YYYY-MM-DD'), 'bob@example.c
Error report -
ORA-20002: Student age is not within the allowed range
ORA-06512: at "COMP214_M23_YIN_8.TRG_STUDENT_AGE_VALIDATION", line 9
ORA-04088: error during execution of trigger 'COMP214_M23_YIN_8.TRG_STUDENT_AGE_VALIDATION
```

```
INSERT INTO Student (student_id, student_name, date_of_birth, email, phone_number)
VALUES (student_id_seq.NEXTVAL, 'Alice', TO_DATE('1960-01-15', 'YYYY-MM-DD'), 'alice@example.com', '123-456-7890');
```



```
Error starting at line : 394 in command -
INSERT INTO Student (student_id, student_name, date_of_birth, email, phone_number)
VALUES (student_id_seq.NEXTVAL, 'Alice', TO_DATE('1960-01-15', 'YYYY-MM-DD'), 'alice@examp
Error report -
ORA-20002: Student age is not within the allowed range
ORA-06512: at "COMP214_M23_YIN_8.TRG_STUDENT_AGE_VALIDATION", line 9
ORA-04088: error during execution of trigger 'COMP214_M23_YIN_8.TRG_STUDENT_AGE_VALIDATION
```

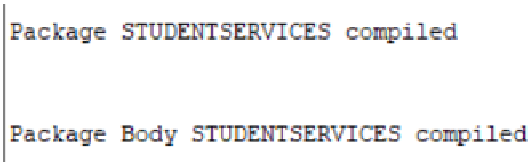
-- 5. Package

```
CREATE OR REPLACE PACKAGE StudentServices AS

    -- Procedures
    PROCEDURE enroll_student(course_id IN NUMBER, student_id IN NUMBER);
    PROCEDURE update_student_email(student_id IN NUMBER, new_email IN VARCHAR2);
    PROCEDURE delete_enrollment(enrollment_id IN NUMBER);

    -- Functions
    FUNCTION get_course_fee(course_id IN NUMBER) RETURN NUMBER;
    FUNCTION calculate_total_fee(student_id IN NUMBER) RETURN NUMBER;
    FUNCTION get_enrolled_students(course_id IN NUMBER) RETURN SYS_REFCURSOR;

    -- Exception
    custom_exception EXCEPTION;

END StudentServices;
/

CREATE OR REPLACE PACKAGE BODY StudentServices AS

    -- Procedures
    PROCEDURE enroll_student(course_id IN NUMBER, student_id IN NUMBER) IS
    BEGIN
        INSERT INTO Enrollment (enrollment_id, student_id, course_id, enrollment_date)
        VALUES (enrollment_id_seq.NEXTVAL, student_id, course_id, SYSDATE);

        COMMIT;
    EXCEPTION
        WHEN OTHERS THEN
            ROLLBACK;
            RAISE;
    END enroll_student;

    PROCEDURE update_student_email(student_id IN NUMBER, new_email IN VARCHAR2) IS
```

```
BEGIN
   UPDATE Student SET email = new_email WHERE student_id = student_id;

   COMMIT;
EXCEPTION
   WHEN OTHERS THEN
      ROLLBACK;
      RAISE;
END update_student_email;

PROCEDURE delete_enrollment(enrollment_id IN NUMBER) IS
BEGIN
   DELETE FROM Enrollment WHERE enrollment_id = enrollment_id;

   COMMIT;
EXCEPTION
   WHEN OTHERS THEN
      ROLLBACK;
      RAISE;
END delete_enrollment;

-- Functions
FUNCTION get_course_fee(course_id IN NUMBER) RETURN NUMBER IS
   course_fee NUMBER;
BEGIN
   SELECT course_fee INTO course_fee FROM Course WHERE course_id = course_id;
   RETURN course_fee;
EXCEPTION
   WHEN NO_DATA_FOUND THEN
```

```
        RETURN NULL;
END get_course_fee;

FUNCTION calculate_total_fee(student_id IN NUMBER) RETURN NUMBER IS
    total_fee NUMBER := 0;
    cursor_enrollments SYS_REFCURSOR;
    enrollment_record Enrollment%ROWTYPE;
BEGIN
    OPEN cursor_enrollments FOR
        SELECT * FROM Enrollment WHERE student_id = student_id;

    LOOP
        FETCH cursor_enrollments INTO enrollment_record;
        EXIT WHEN cursor_enrollments%NOTFOUND;

        total_fee := total_fee + get_course_fee(enrollment_record.course_id);
    END LOOP;

    CLOSE cursor_enrollments;

    RETURN total_fee;
EXCEPTION
    WHEN OTHERS THEN
        RETURN NULL;
END calculate_total_fee;

FUNCTION get_enrolled_students(course_id IN NUMBER) RETURN SYS_REFCURSOR IS
    enrolled_students SYS_REFCURSOR;
BEGIN
```

```
    OPEN enrolled_students FOR
      SELECT s.student_id, s.student_name
      FROM Student s
      JOIN Enrollment e ON s.student_id = e.student_id
      WHERE e.course_id = course_id;

    RETURN enrolled_students;
  END get_enrolled_students;

END StudentServices;
/
```



```
Package STUDENTSERVICES compiled


Package Body STUDENTSERVICES compiled
```