Week 2 - Assignment

Created by	DG Dakshil Gorasiya

Assignment 6

```
-- 1. Find students who scored above the average score using a subquery.
SELECT DISTINCT
    T01F01,
    T01F02
FROM
    T01
    INNER JOIN T03 ON T01F01 = T03F02
WHERE
    T03F04 > (SELECT AVG(T03F04) FROM T03);


-- 2. Get students enrolled in the same course as “John” using a correlated subquery.
SELECT
    T01F01,
    T01F02
FROM
    T01
    INNER JOIN T03 ON T01F01 = T03F02
WHERE
    T03F03 = (SELECT T03F03 FROM T03 INNER JOIN T01 ON T03F02 = T01F01 WHERE T01F02 = 'Alice Johnson');
```

Assignment 7

```
-- 1. Write a query to list all distinct course names from Courses and Marks tables (use UNION).
SELECT
    T02F02
FROM
    T02
UNION
SELECT
    T03F03
FROM
    T03;

-- 2. Write another query to include duplicates (use UNION ALL).
SELECT
    T02F02
FROM
    T02
UNION ALL
SELECT
    T03F03
```

```
FROM
    T03;
```

Assignment 8

```
-- 1. Add a PRIMARY KEY on student_id.
ALTER TABLE T01 DROP PRIMARY KEY;
ALTER TABLE T01 ADD CONSTRAINT PK_T01F01 PRIMARY KEY(T01F01);

-- 2. Add an AUTO_INCREMENT to course_id.
ALTER TABLE T01 DROP CONSTRAINT T01F05_FK; -- drop foreign key to add AUTO_INCREMENT in t02
ALTER TABLE T02 MODIFY COLUMN T02F01 INT AUTO_INCREMENT;
ALTER TABLE T01 ADD CONSTRAINT T01F05_FK FOREIGN KEY (T01F05) REFERENCES T02(T02F01);

INSERT INTO T02 (T02F02, T02F03) VALUES
('Backend development', 4);

SELECT * FROM T02;

ALTER TABLE T01 DROP CONSTRAINT T01F06; -- drop old unique contraint
-- 3. Create an INDEX on email for faster search.
CREATE UNIQUE INDEX INDEX_EMAIL ON T01(T01F06);



-- 4. Prove query optimization difference using EXPLAIN with and without index.
EXPLAIN SELECT
    T01F01,
    T01F02,
    T01F06
FROM
    T01
WHERE
    T01F06 = 'EVE@GMAIL.COM';
```

Before index

id	select_type	table	partitions	type	possible_keys	key	key_len
1	SIMPLE	T01		ALL			

After index

id	select_type	table	partitions	type	possible_keys	key	key_len
1	SIMPLE	T01		const	INDEX_EMAIL	INDEX_EMAIL	402

Assignment 9

```
-- 1. Write a Stored Procedure to return all students enrolled in a given course.
DROP PROCEDURE IF EXISTS GET_STUDENT_ENROLLED_IN_COURSE;
```

```
DELIMITER $$
CREATE PROCEDURE GET_STUDENT_ENROLLED_IN_COURSE(IN p_course_name VARCHAR(50))
BEGIN
    SELECT
        T01F01,
        T01F02
    FROM
        T01
        INNER JOIN T02 ON T01F05 = T02F01
    WHERE
        T02F02 = p_course_name;
END$$
DELIMITER ;

CALL GET_STUDENT_ENROLLED_IN_COURSE("Business Administration");

-- 2. Create a Function to calculate grade based on marks (e.g., A/B/C).
DELIMITER $$
CREATE FUNCTION CALCULATE_GRADE(p_marks DECIMAL(5,2))
RETURNS CHAR
DETERMINISTIC
BEGIN
    DECLARE v_grade CHAR(1);

    IF p_marks >= 80 THEN
        SET v_grade = 'A';
    ELSEIF p_marks >=  60 THEN
        SET v_grade = 'B';
    ELSEIF p_marks >= 40 THEN
        SET v_grade = 'C';
    ELSE
        SET v_grade = 'F';
    END IF;

    RETURN v_grade;
END$$
DELIMITER ;

SELECT
    CALCULATE_GRADE(70);


-- 3. Create a Trigger to log deleted student records into a new table DeletedStudents.
# Delted student
DROP TABLE IF EXISTS T04;
CREATE TABLE T04
(
    T04F01 INT NOT NULL, # student id
    T04F02 VARCHAR(30) NOT NULL, # name
    T04F03 INT NOT NULL, # age
    T04F04 ENUM("MALE", "FEMALE", "OTHER") NOT NULL, # gender
    T04F05 VARCHAR(100) NOT NULL, # email
    T04F06 DATE DEFAULT (CURDATE()) # deletion date
);


DROP TRIGGER IF EXISTS AFTER_STUDENT_DELETE;
```

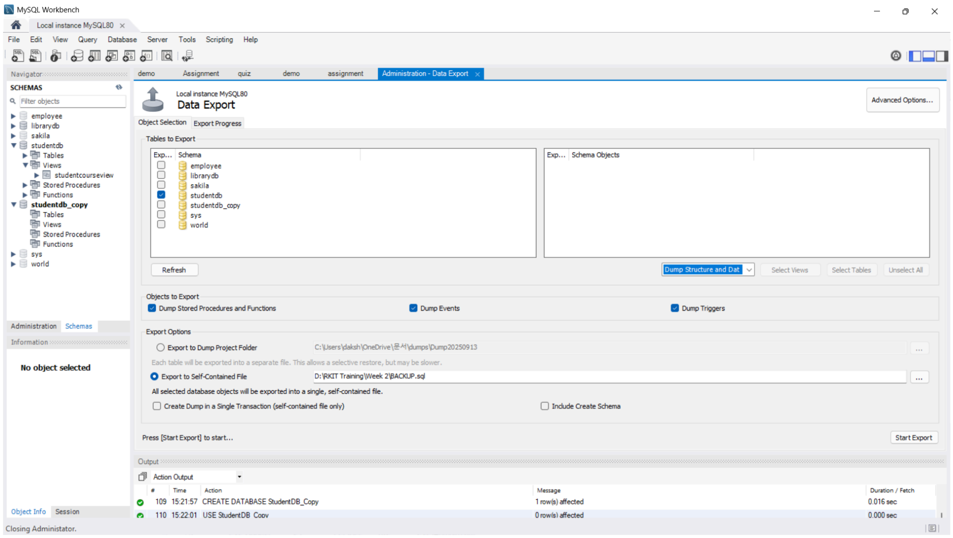
```
DELIMITER $$
CREATE TRIGGER AFTER_STUDENT_DELETE
AFTER DELETE ON T01
FOR EACH ROW
BEGIN
    INSERT INTO T04(T04F01, T04F02, T04F03, T04F04, T04F05) VALUES
    (OLD.T01F01, OLD.T01F02, OLD.T01F03, OLD.T01F04, OLD.T01F06);
END$$
DELIMITER ;

SELECT * FROM T01;
DELETE FROM T01 WHERE T01F01 = 2;
SELECT * FROM T04;
```

Assignment 10

```
-- 1. Create a View StudentCourseView that shows student name + course name.
CREATE OR REPLACE VIEW STUDENTCOURSEVIEW AS
SELECT
    T01F01,
    T01F02,
    T02F02
FROM
    T01
    INNER JOIN T02 ON T01F05 = T02F01;

-- 2. Query from the view to find students enrolled in “Database Systems”.
SELECT
    T01F01,
    T01F02,
    T02F02
FROM
    STUDENTCOURSEVIEW
WHERE
    T02F02 = "Computer Science";
```



Local instance MySQL80
Data Export

Tables to Export

Exp...	Schema
□	employee
□	librarydb
□	sakila
☑	studentdb
□	studentdb_copy
□	sys
□	world

Exp... Schema Objects

Dump Structure and Dat

Objects to Export

☑ Dump Stored Procedures and Functions ☑ Dump Events ☑ Dump Triggers

Export Options

○ Export to Dump Project Folder C:\Users\daksh\OneDrive\문서\dumps\Dump20250913

Each table will be exported into a separate file. This allows a selective restore, but may be slower.

● Export to Self-Contained File D:\RKIT Training\Week 2\BACKUP.sql

All selected database objects will be exported into a single, self-contained file.

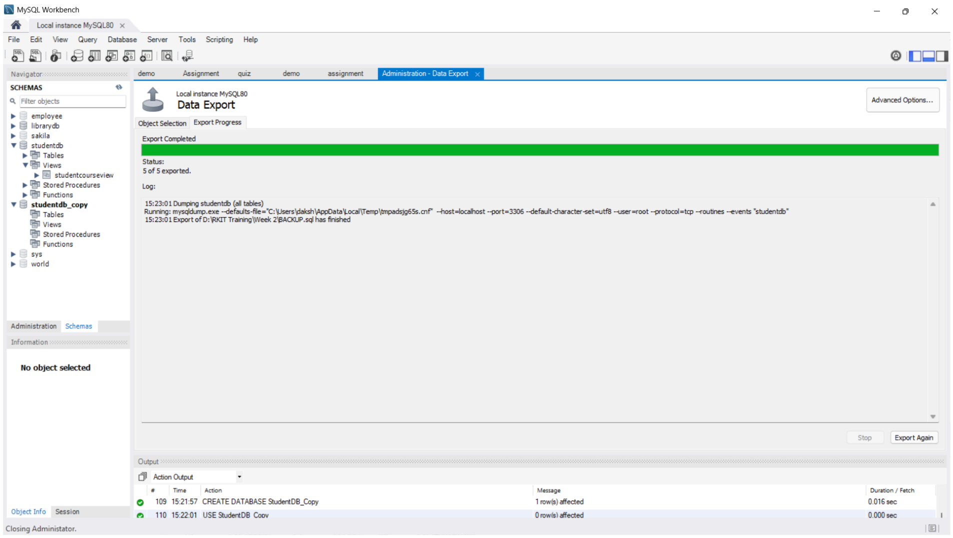
□ Create Dump in a Single Transaction (self-contained file only) □ Include Create Schema

Press [Start Export] to start...

	#	Time	Action	Message	Duration / Fetch
✓	109	15:21:57	CREATE DATABASE StudentDB_Copy	1 row(s) affected	0.016 sec
✓	110	15:22:01	USE StudentDB_Copy	0 row(s) affected	0.000 sec

No object selected

Closing Administator.



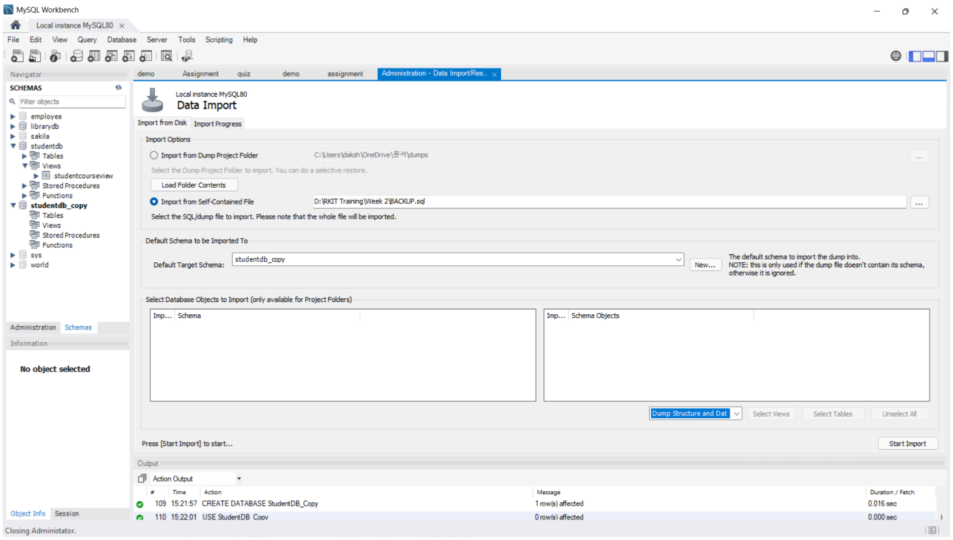
Export Completed

Status:
5 of 5 exported.

Log:

```
15:23:01 Dumping studentdb (all tables)
Running: mysqldump.exe --defaults-file="C:\Users\daksh\AppData\Local\Temp\tmpadsjg65s.cnf"  --host=localhost --port=3306 --default-character-set=utf8 --user=root --protocol=tcp --routines --events "studentdb"
15:23:01 Export of D:\RKIT Training\Week 2\BACKUP.sql has finished
```

	#	Time	Action	Message	Duration / Fetch
✓	109	15:21:57	CREATE DATABASE StudentDB_Copy	1 row(s) affected	0.016 sec
✓	110	15:22:01	USE StudentDB_Copy	0 row(s) affected	0.000 sec



Import Options

○ Import from Dump Project Folder C:\Users\daksh\OneDrive\문서\dumps

Select the Dump Project Folder to import. You can do a selective restore.

● Import from Self-Contained File D:\RKIT Training\Week 2\BACKUP.sql

Select the SQL/dump file to import. Please note that the whole file will be imported.

Default Schema to be Imported To

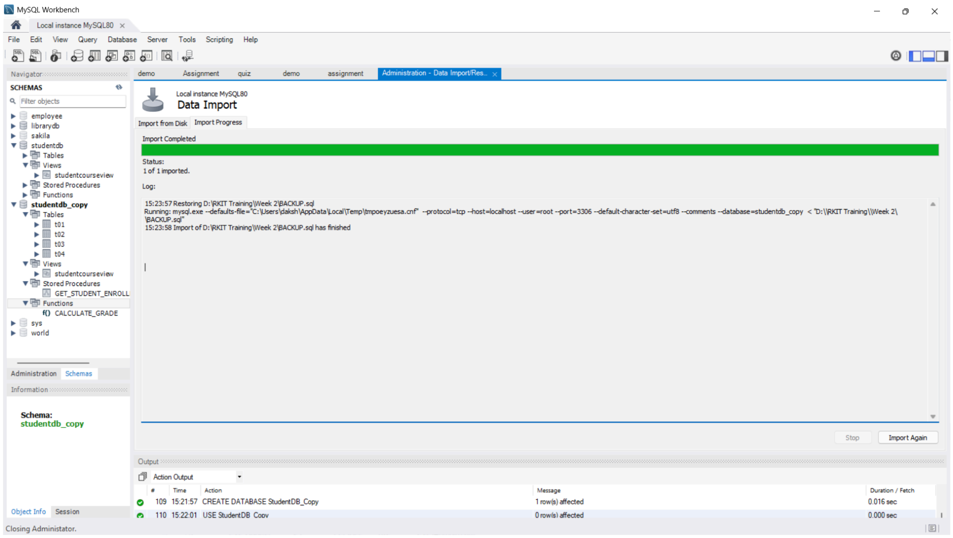
Default Target Schema: studentdb_copy

The default schema to import the dump into.
NOTE: this is only used if the dump file doesn't contain its schema, otherwise it is ignored.

Select Database Objects to Import (only available for Project Folders)

Press [Start Import] to start...

	#	Time	Action	Message	Duration / Fetch
✓	109	15:21:57	CREATE DATABASE StudentDB_Copy	1 row(s) affected	0.016 sec
✓	110	15:22:01	USE StudentDB_Copy	0 row(s) affected	0.000 sec



Schema:
studentdb_copy

Import Completed

Status:
1 of 1 imported.

Log:

```
15:23:57 Restoring D:\RKIT Training\Week 2\BACKUP.sql
Running: mysql.exe --defaults-file="C:\Users\daksh\AppData\Local\Temp\tmpoeyzuesa.cnf"  --protocol=tcp --host=localhost --user=root --port=3306 --default-character-set=utf8 --comments --database=studentdb_copy  < "D:\\RKIT Training\\Week 2\
\BACKUP.sql"
15:23:58 Import of D:\RKIT Training\Week 2\BACKUP.sql has finished
```

	#	Time	Action	Message	Duration / Fetch
✓	109	15:21:57	CREATE DATABASE StudentDB_Copy	1 row(s) affected	0.016 sec
✓	110	15:22:01	USE StudentDB_Copy	0 row(s) affected	0.000 sec