

Academics Management System [Task 1] – Results Images

1. Database Creation:

a) Create the StudentInfo table with columns STU_ID, STU_NAME, DOB, PHONE_NO, EMAIL_ID, ADDRESS.

b) Create the CoursesInfo table with columns COURSE_ID, COURSE_NAME, COURSE_INSTRUCTOR NAME.

c) Create the EnrollmentInfo with columns ENROLLMENT_ID, STU_ID, COURSE_ID, ENROLL_STATUS(Enrolled/Not Enrolled).

1. Data Creation:

Insert some sample data for StudentInfo table , CoursesInfo table, EnrollmentInfo with respective fields.

**All insert and create operations done -
Attaching results image –**

The screenshot displays the SQL Server Enterprise Manager interface. The top pane shows a SQL script with the following queries:

```
20 COURSE_ID INT,  
21 ENROLL_STATUS VARCHAR(50),  
22 FOREIGN KEY (STU_ID) REFERENCES StudentInfo(STU_ID),  
23 FOREIGN KEY (COURSE_ID) REFERENCES CoursesInfo(COURSE_ID)  
24 );  
25  
26  
27  
28 INSERT INTO StudentInfo (STU_ID, STU_NAME, DOB, PHONE_NO, EMAIL_ID, ADDRESS)  
29 VALUES  
30 (1, 'Bruce Wayne', '1980-05-27', '123-456-7890', 'bruce.wayne@waynecorp.com', 'Wayne Manor, Gotham');
```

The bottom pane shows the 'Output' window with the following results:

#	Time	Action	Message	Duration / Fetch
1	21:42:01	CREATE TABLE StudentInfo (STU_ID INT PRIMARY KEY, STU_NAME VARC...	0 row(s) affected	0.047 sec
2	21:42:01	CREATE TABLE CoursesInfo (COURSE_ID INT PRIMARY KEY, COURSE_NA...	0 row(s) affected	0.031 sec
3	21:42:01	CREATE TABLE EnrollmentInfo (ENROLLMENT_ID INT PRIMARY KEY, STU_I...	0 row(s) affected	0.047 sec
4	21:42:01	INSERT INTO StudentInfo (STU_ID, STU_NAME, DOB, PHONE_NO, EMAIL_ID, AD...	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0	0.000 sec
5	21:42:01	INSERT INTO CoursesInfo (COURSE_ID, COURSE_NAME, COURSE_INSTRUCTO...	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0	0.015 sec
6	21:42:01	INSERT INTO EnrollmentInfo (ENROLLMENT_ID, STU_ID, COURSE_ID, ENROLL...	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.000 sec

3) Retrieve the Student Information

a) Write a query to retrieve student details, such as student name, contact informations, and Enrollment status.

```

Query 1  SQL File 1*  SQL File 2*  Administration - Server Status
Limit to 1000 rows
1 • SELECT STU_NAME, PHONE_NO, EMAIL_ID, ENROLL_STATUS
2 FROM StudentInfo SI
3 JOIN EnrollmentInfo EI ON SI.STU_ID = EI.STU_ID;

```

<				
Result Grid				
Filter Rows: <input type="text"/>				
Export:				
Wrap Cell Content:				
	STU_NAME	PHONE_NO	EMAIL_ID	ENROLL_STATUS
▶	Bruce Wayne	123-456-7890	bruce.wayne@waynecorp.com	Enrolled
	Bruce Wayne	123-456-7890	bruce.wayne@waynecorp.com	Enrolled
	Clark Kent	987-654-3210	clark.kent@dailypplanet.com	Enrolled
	Clark Kent	987-654-3210	clark.kent@dailypplanet.com	Not Enrolled
	Diana Prince	555-666-7777	diana.prince@themiscira.com	Enrolled
	Diana Prince	555-666-7777	diana.prince@themiscira.com	Enrolled
	Barry Allen	333-444-5555	barry.allen@ccpd.com	Enrolled
	Barry Allen	333-444-5555	barry.allen@ccpd.com	Enrolled
	Arthur Curry	222-333-4444	arthur.curry@atlantis.com	Enrolled
	Arthur Curry	222-333-4444	arthur.curry@atlantis.com	Not Enrolled

b) Write a query to retrieve a list of courses in which a specific student is enrolled.

Query 1 SQL File 1* **SQL File 2*** Administration - Server Status

Limit to 1000 rows

```

1 • SELECT C.COURSE_NAME, C.COURSE_INSTRUCTOR_NAME
2 FROM EnrollmentInfo E
3 JOIN CoursesInfo C ON E.COURSE_ID = C.COURSE_ID
4 WHERE E.STU_ID = (SELECT STU_ID FROM StudentInfo WHERE STU_NAME = 'Bruce Wayne');

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: [IA](#)

	COURSE_NAME	COURSE_INSTRUCTOR_NAME
▶	Combat Training	Oliver Queen
	Mythology	Hippolyta

c) Write a query to retrieve course information, including course name, instructor information.

Query 1 SQL File 1* **SQL File 2*** Administration - Server Status

Limit to 1000 rows

```

1 • SELECT COURSE_NAME, COURSE_INSTRUCTOR_NAME
2 FROM CoursesInfo;

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: [IA](#)

	COURSE_NAME	COURSE_INSTRUCTOR_NAME
▶	Combat Training	Oliver Queen
	Advanced Physics	Ray Palmer
	Mythology	Hippolyta
	Forensics	Iris West
	Marine Biology	Mera

d) Write a query to retrieve course information for a specific course .

Query 1 SQL File 1* **SQL File 2*** × Administration - Server Status

Limit to 1000 rows

```

1 • SELECT COURSE_NAME, COURSE_INSTRUCTOR_NAME
2   FROM CoursesInfo
3  WHERE COURSE_NAME = 'Combat Training';

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

	COURSE_NAME	COURSE_INSTRUCTOR_NAME
▶	Combat Training	Oliver Queen

e) Write a query to retrieve course information for multiple courses.

Query 1 SQL File 1* **SQL File 2*** × Administration - Server Status

Limit to 1000 rows

```

1 • SELECT COURSE_NAME, COURSE_INSTRUCTOR_NAME
2   FROM CoursesInfo
3  WHERE COURSE_NAME IN ('Combat Training', 'Mythology');

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

	COURSE_NAME	COURSE_INSTRUCTOR_NAME
▶	Combat Training	Oliver Queen
	Mythology	Hippolyta

f) Test the queries to ensure accurate retrieval of student information.(execute the queries andverify the results against the expected output.)

4. Reporting and Analytics (Using joining queries)

a) Write a query to retrieve the number of students enrolled in each course

Query 1 SQL File 1* SQL File 2* Administration - Server Status

Limit to 1000 rows

```

1 • SELECT C.COURSE_NAME, COUNT(E.STU_ID) AS NumberOfStudents
2 FROM CoursesInfo C
3 LEFT JOIN EnrollmentInfo E ON C.COURSE_ID = E.COURSE_ID
4 GROUP BY C.COURSE_NAME;

```

Result Grid Filter Rows: Export: Wrap Cell Content: [IA](#)

	COURSE_NAME	NumberOfStudents
▶	Combat Training	3
	Advanced Physics	3
	Mythology	2
	Forensics	1
	Marine Biology	1

b) Write a query to retrieve the list of students enrolled in a specific course

Query 1 SQL File 1* SQL File 2* Administration - Server Status

Limit to 1000 rows

```

1 • SELECT S.STU_NAME
2 FROM StudentInfo S
3 JOIN EnrollmentInfo E ON S.STU_ID = E.STU_ID
4 WHERE E.COURSE_ID = (SELECT COURSE_ID FROM CoursesInfo WHERE COURSE_NAME = 'Combat Training');

```

Result Grid Filter Rows: Export: Wrap Cell Content: [IA](#)

	STU_NAME
▶	Bruce Wayne
	Clark Kent
	Barry Allen

c) Write a query to retrieve the count of enrolled students for each instructor.

Query 1 SQL File 1* SQL File 2* Administration - Server Status

Limit to 1000 rows

```

1 • SELECT C.COURSE_INSTRUCTOR_NAME, COUNT(E.STU_ID) AS NumberOfStudents
2 FROM CoursesInfo C
3 LEFT JOIN EnrollmentInfo E ON C.COURSE_ID = E.COURSE_ID
4 GROUP BY C.COURSE_INSTRUCTOR_NAME;

```

Result Grid Filter Rows: Export: Wrap Cell Content:

COURSE_INSTRUCTOR_NAME	NumberOfStudents
Oliver Queen	3
Ray Palmer	3
Hippolyta	2
Iris West	1
Mera	1

d) Write a query to retrieve the list of students who are enrolled in multiple courses

Query 1 SQL File 1* SQL File 2* Administration - Server Status

Limit to 1000 rows

```

1 • SELECT S.STU_NAME, COUNT(E.COURSE_ID) AS NumberOfCourses
2 FROM StudentInfo S
3 JOIN EnrollmentInfo E ON S.STU_ID = E.STU_ID
4 GROUP BY S.STU_NAME
5 HAVING COUNT(E.COURSE_ID) > 1;

```

Result Grid Filter Rows: Export: Wrap Cell Content:

STU_NAME	NumberOfCourses
Bruce Wayne	2
Clark Kent	2
Diana Prince	2
Barry Allen	2
Arthur Curry	2

e) Write a query to retrieve the courses that have the highest number of enrolled students(arranging from highest to lowest)

Query 1 SQL File 1* SQL File 2* x Administration - Server Status

Limit to 1000 rows

```
1 • SELECT C.COURSE_NAME, COUNT(E.STU_ID) AS NumberOfStudents
2 FROM CoursesInfo C
3 LEFT JOIN EnrollmentInfo E ON C.COURSE_ID = E.COURSE_ID
4 GROUP BY C.COURSE_NAME
5 ORDER BY NumberOfStudents DESC;
```

<

Result Grid Filter Rows: Export: Wrap Cell Content: [IA](#)

	COURSE_NAME	NumberOfStudents
▶	Combat Training	3
	Advanced Physics	3
	Mythology	2
	Forensics	1
	Marine Biology	1