Introduction to Data Science

Homework Assignment-3

Kirti Katiyar

Problem 1: Find two tables on the same topic from Web1 that have some attributes in common (e.g., city and salary as in the example below). Write a SQL query fusing these tables, so that the query result set has combined information from both tables, which yields a quantitatively enriched dataset.

a. Each pair of tables that you have found (the attribute names and 5 sample rows from each table)
b. The SQL query fusing these two tables and the result set of running the query on your tables (no more than 5 first rows)

Pair 1:

a. Table 1- Reliance_Digital

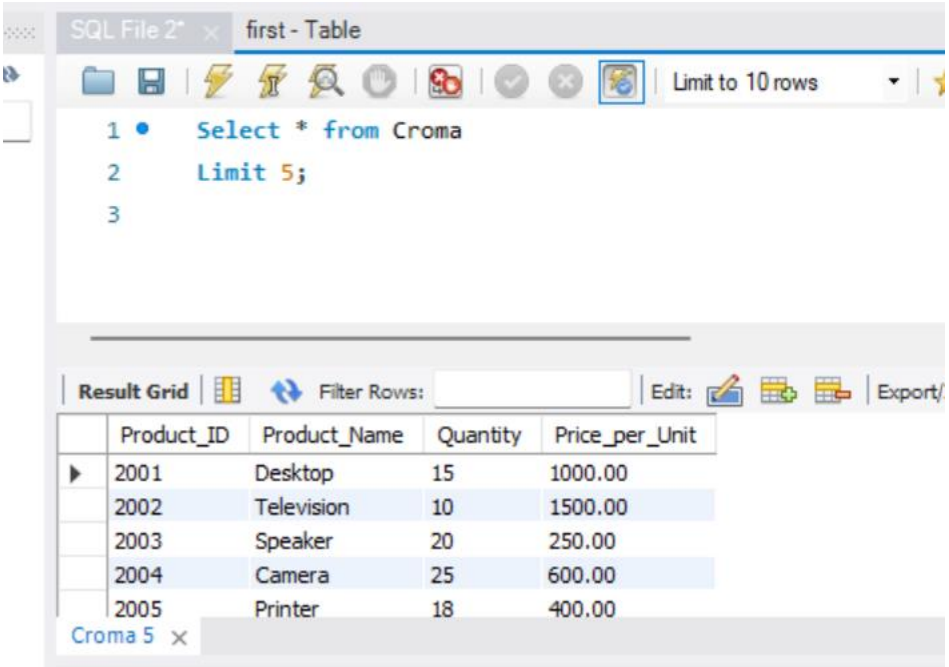
```
select * from Reliance_Digital
LIMIT 5;
```

Result Grid | Filter Rows: | Edit: | Export/Imp

Product_ID	Product_Name	Quantity	Price_per_Unit
1001	Laptop	20	1200.00
1002	Smartphone	30	800.00
1003	Tablet	25	500.00
1004	Headphones	40	200.00
1005	Smartwatch	35	300.00

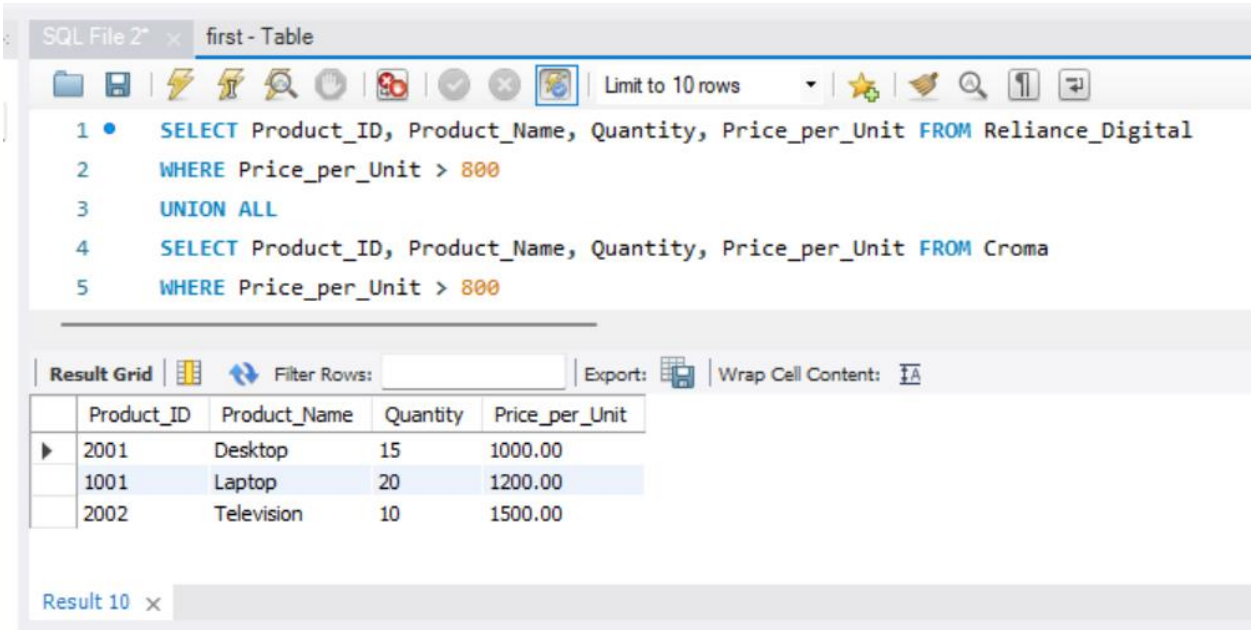
Reliance_Digital 4

Table 2- Croma



Product_ID	Product_Name	Quantity	Price_per_Unit
2001	Desktop	15	1000.00
2002	Television	10	1500.00
2003	Speaker	20	250.00
2004	Camera	25	600.00
2005	Printer	18	400.00

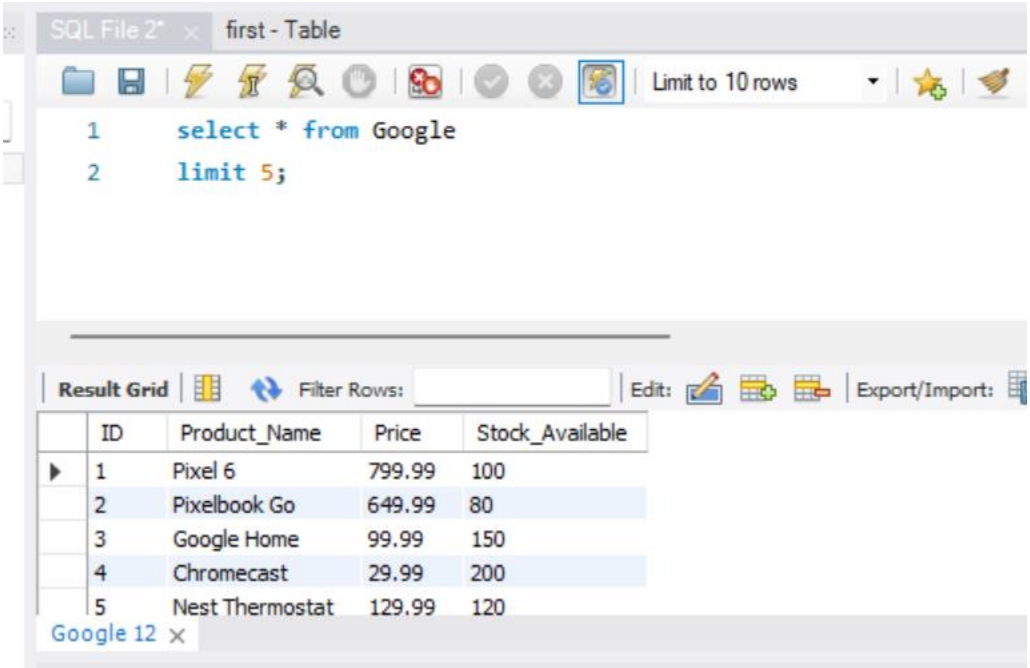
SQL query



Product_ID	Product_Name	Quantity	Price_per_Unit
2001	Desktop	15	1000.00
1001	Laptop	20	1200.00
2002	Television	10	1500.00

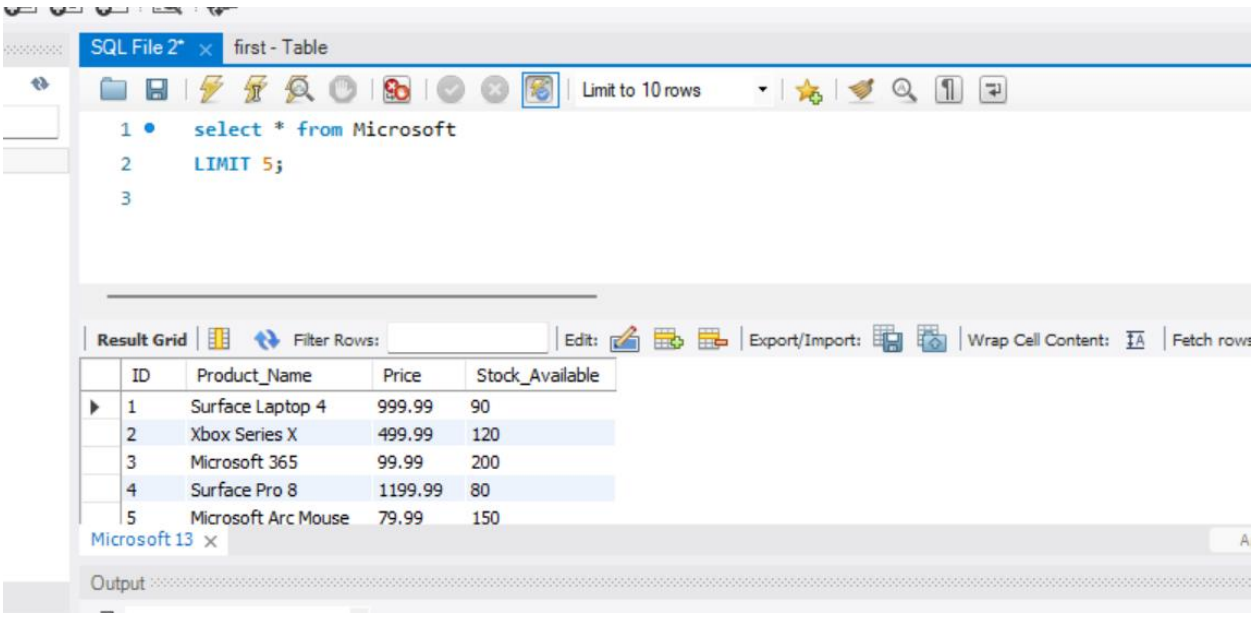
Pair 2:

a. **Table 1- Google**



ID	Product_Name	Price	Stock_Available
1	Pixel 6	799.99	100
2	Pixelbook Go	649.99	80
3	Google Home	99.99	150
4	Chromecast	29.99	200
5	Nest Thermostat	129.99	120

Table 2- Microsoft



ID	Product_Name	Price	Stock_Available
1	Surface Laptop 4	999.99	90
2	Xbox Series X	499.99	120
3	Microsoft 365	99.99	200
4	Surface Pro 8	1199.99	80
5	Microsoft Arc Mouse	79.99	150

SQL Query

SQL File 2* | first - Table

Limit to 10 rows

```sql
SELECT Product_Name, Price, Stock_Available, 'Google' AS Company FROM Google
WHERE Stock_Available > 120
UNION ALL
SELECT Product_Name, Price, Stock_Available, 'Microsoft' AS Company FROM Microsoft
WHERE Stock_Available > 120
ORDER BY Price DESC;
```

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

Product_Name	Price	Stock_Available	Company
Google Home	99.99	150	Google
Microsoft 365	99.99	200	Microsoft
Microsoft Arc Mouse	79.99	150	Microsoft
Chromecast	29.99	200	Google

Result 18

Output

Pair 3:

Table 1- Facebook

SQL File 2* | first - Table

Limit to 10 rows

```sql
select * from Facebook
limit 5;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: | Fetch rows:

ID	User_Name	Posts	Followers	Following	Engagement_Rate	Category	Hashtags	Average_
1	John Doe	300	20000	300	3.20	Lifestyle	#travel #food #photography	150
2	Jane Smith	500	30000	400	5.10	Fashion	#style #beauty #ootd	250
3	Michael Johnson	400	25000	350	4.00	Photography	#nature #landscape #art	180
4	Emily Williams	600	18000	250	4.80	Art	#painting #drawing #creativity	220
5	Robert Brown	800	35000	500	6.20	Design	#graphicdesign #illustration #branding	300

Output

Action Output

Table 2- Instagram

SQL File 2* | first - Table

Limit to 10 rows

```sql
select * from Instagram
limit 5;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: | Fetch rows:

ID	User_Name	Posts	Followers	Following	Engagement_Rate	Category	Hashtags	Avera
1	johndoe_insta	500	15000	200	4.50	Lifestyle	#travel #food #photography	200
2	janesmith_ig	300	20000	300	3.20	Fashion	#style #beauty #ootd	150
3	mjohnson_ig	800	30000	400	5.10	Photography	#nature #landscape #art	250
4	emilywilliams.official	400	25000	350	4.00	Art	#painting #drawing #creativity	180
5	robertbrown_designs	600	18000	250	4.80	Design	#graphicdesign #illustration #branding	220

a. SQL Query

SQL File 2* | first - Table

Limit to 10 rows

```sql
SELECT User_Name, Posts, Followers, Following, Engagement_Rate,
Category, Hashtags, Average_Likes, 'Facebook' AS Platform FROM Facebook
WHERE Category LIKE '%Photography%' OR Followers > 25000
UNION ALL
SELECT User_Name, Posts, Followers, Following, Engagement_Rate, Category,
Hashtags, Average_Likes, 'Instagram' AS Platform FROM Instagram
WHERE Category LIKE '%Photography%' OR Followers > 25000
ORDER BY User_Name;
```

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

User_Name	Posts	Followers	Following	Engagement_Rate	Category	Hashtags	Average_Likes
Jane Smith	500	30000	400	5.10	Fashion	#style #beauty #ootd	250
Michael Johnson	400	25000	350	4.00	Photography	#nature #landscape #art	180
mjohnson_ig	800	30000	400	5.10	Photography	#nature #landscape #art	250
Robert Brown	800	35000	500	6.20	Design	#graphicdesign #illustration #branding	300

Result 25 | Read Only

Pair 4:

Table 1- Newyork

SQL File 2* | first - Table

Limit to 10 rows

```
select * from NewYork
limit 5;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: | Fetch rows:

ID	City	Population	Area_SqMiles	Median_Income	Crime_Rate	Median_Rent	Education_Rank	Unemployment_Rate
1	New York City	8500000	468.90	60000.00	3.50	2000.00	8	5.10
2	Brooklyn	2700000	97.97	55000.00	4.20	1800.00	7	6.20
3	Queens	2300000	108.53	58000.00	3.90	1900.00	7	5.80
4	Bronx	1400000	42.10	45000.00	5.10	1500.00	5	8.30
5	Staten Island	500000	58.37	70000.00	2.70	2200.00	9	4.90

Table 2- Chicago

SQL File 2* | first - Table

Limit to 10 rows

```
Select * From Chicago
Limit 5;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: | Fetch rows:

ID	City	Population	Area_SqMiles	Median_Income	Crime_Rate	Median_Rent	Education_Rank	Unemployment_Rate
1	Chicago	2700000	227.63	55000.00	4.50	1500.00	6	6.20
2	Aurora	200000	45.80	50000.00	3.80	1200.00	5	7.50
3	Naperville	150000	39.32	75000.00	2.10	1800.00	9	4.20
4	Joliet	120000	62.89	48000.00	5.30	1100.00	4	8.90
5	Schaumburg	80000	19.33	60000.00	3.40	1600.00	7	5.60

SQL Query

SQL File 2* | first - Table

Limit to 10 rows

```sql
SELECT City, Population, Area_SqMiles, Median_Income, Crime_Rate, Median_Rent,
Education_Rank, Unemployment_Rate, 'New York' AS Location FROM NewYork
WHERE Population > 1000000 AND (Crime_Rate < 4 OR Median_Rent < 2000)
UNION ALL
SELECT City, Population, Area_SqMiles, Median_Income, Crime_Rate, Median_Rent,
Education_Rank, Unemployment_Rate, 'Chicago' AS Location FROM Chicago
WHERE Population > 500000 AND (Crime_Rate < 4 OR Median_Rent < 2000)
ORDER BY City;
```

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

City	Population	Area_SqMiles	Median_Income	Crime_Rate	Median_Rent	Education_Rank	Unemployment_Rate	Locatio
Bronx	1400000	42.10	45000.00	5.10	1500.00	5	8.30	New Yor
Brooklyn	2700000	97.97	55000.00	4.20	1800.00	7	6.20	New Yor
Chicago	2700000	227.63	55000.00	4.50	1500.00	6	6.20	Chicago
New York City	8500000	468.90	60000.00	3.50	2000.00	8	5.10	New Yor

Result 37 | Read Only

Pair: 5

Table 1- SoftwareEngineer

SQL File 2* | first - Table

Limit to 10 rows

```sql
select * from SoftwareEngineer
LIMIT 5;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: | Fetch rows:

ID	Name	Age	ExperienceYears	ProgrammingLanguages	Company	Salary	Location	Degree
1	John Doe	30	8	Java, Python, C++	Google	150000.00	California	Computer Science
2	Jane Smith	28	5	JavaScript, C#, SQL	Microsoft	130000.00	Washington	Software Engineering
3	Michael Johnson	35	10	Ruby, Swift, PHP	Apple	160000.00	California	Computer Science
4	Emily Williams	32	6	Python, C, Ruby	Facebook	140000.00	California	Computer Engineering
5	Robert Brown	29	7	Java, JavaScript, Python	Amazon	135000.00	Washington	Computer Science

SoftwareEngineer 27 | Apply | Revert | Context Help | Snipp

SQLAdditions

Automatic c
disabled. Use
manually g
current care
toggle aut

Output

Table 2- MechanicalEngineer

SQL File 2* | first - Table

Limit to 10 rows

```sql
SELECT * FROM MechanicalEngineer
LIMIT 5;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: | Fetch rows:

ID	Name	Age	ExperienceYears	Skills	Company	Salary	Location
1	William Johnson	31	8	CAD, SolidWorks, ANSYS	Tesla	145000.00	California
2	Emma Davis	27	5	AutoCAD, MATLAB, CFD	Boeing	120000.00	Washington
3	Michael Garcia	33	10	Thermodynamics, HVAC, FEA	General Motors	155000.00	Michigan
4	Olivia Wilson	29	6	Robotics, Mechatronics, Heat Transfer	Lockheed Martin	130000.00	Maryland
5	Ethan Anderson	28	7	Fluid Mechanics, Material Science, Machine Design	SpaceX	135000.00	California

a. SQL Query:

SQL File 2* | first - Table

Limit to 10 rows

```sql
SELECT Name, Age, ExperienceYears, Company, Salary, Location, Degree,
'Software Engineer' AS Field FROM SoftwareEngineer
WHERE ExperienceYears > 6 AND (Location = 'California' OR Location = 'Washington')
UNION ALL
SELECT Name, Age, ExperienceYears, Company, Salary, Location, Degree,
 'Mechanical Engineer' AS Field FROM MechanicalEngineer
WHERE ExperienceYears > 6 AND (Location = 'California' OR Location = 'Washington')
ORDER BY Name;
```

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

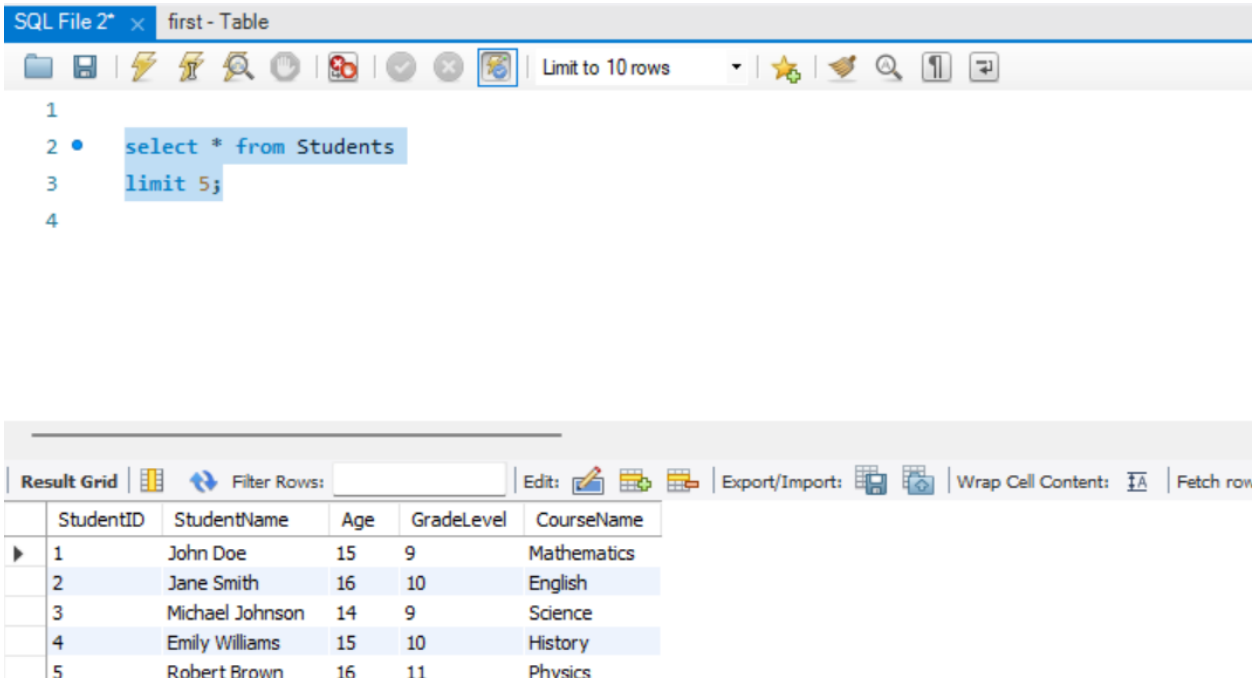
Name	Age	ExperienceYears	Company	Salary	Location	Degree	Field
Ethan Anderson	28	7	SpaceX	135000.00	California	Mechanical Engineering	Mechanical Engineer
John Doe	30	8	Google	150000.00	California	Computer Science	Software Engineer
Michael Johnson	35	10	Apple	160000.00	California	Computer Science	Software Engineer
Robert Brown	29	7	Amazon	135000.00	Washington	Computer Science	Software Engineer
William Johnson	31	8	Tesla	145000.00	California	Mechanical Engineering	Mechanical Engineer

Result 32 | Read Or

Problem 2: Find two tables on the same topic from the Web that have some, but not all attributes in common (e.g. author and title are shared, but not lyrics, genre, and length as in T1 and T2 below). Write a SQL query fusing these tables, so that the query result set has attributes from both tables, which yields a structurally enriched dataset.

Pair 1:

Table 1- Students



StudentID	StudentName	Age	GradeLevel	CourseName
1	John Doe	15	9	Mathematics
2	Jane Smith	16	10	English
3	Michael Johnson	14	9	Science
4	Emily Williams	15	10	History
5	Robert Brown	16	11	Physics

Table 2- Grades

SQL File 2* | first - Table

Limit to 10 rows

```sql
select * from Grades
limit 5;
```

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

StudentID	CourseName	Grade
1	Mathematics	90.5
2	English	85.0
3	Science	92.3
4	History	88.7
5	Physics	94.2

Grades 67

SQL Query:

SQL File 2* | first - Table

Limit to 10 rows

```sql
SELECT s.StudentID, s.StudentName, s.Age, s.GradeLevel, s.CourseName, g.Grade
FROM Students s
JOIN Grades g ON s.CourseName = g.CourseName
WHERE s.StudentName LIKE '%John%' AND s.GradeLevel = 9;
```

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

StudentID	StudentName	Age	GradeLevel	CourseName	Grade
1	John Doe	15	9	Mathematics	90.5
3	Michael Johnson	14	9	Science	92.3

Result 70

Pair 2:

Table 1- movies

SQL File 2* | first - Table

Limit to 10 rows

```sql
select * from movies
limit 5;
```

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

MovieID	MovieTitle	Genre	ReleaseYear	Director
1	The Shawshank Redemption	Drama	1994	Frank Darabont
2	Inception	Sci-Fi	2010	Christopher Nolan
3	Forrest Gump	Drama	1994	Robert Zemeckis
4	The Dark Knight	Action	2008	Christopher Nolan
5	The Matrix	Sci-Fi	1999	The Wachowskis

movies 47

Output

Table 2- actors

SQL File 2* | first - Table

Limit to 10 rows

```sql
select * from actors
limit 5;
```

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

ActorID	MovieTitle	Name	Age	Gender	MovieID
1	The Shawshank Redemption	Tim Robbins	62	Male	1
2	The Shawshank Redemption	Morgan Freeman	84	Male	1
3	Inception	Leonardo DiCaprio	47	Male	2
4	Forrest Gump	Tom Hanks	65	Male	3
5	The Dark Knight	Heath Ledger	28	Male	4

actors 48

SQL Query:

SQL File 2* | first - Table

Limit to 10 rows

```sql
SELECT m.MovieTitle, m.Genre, m.ReleaseYear, m.Director, a.Name, a.Age, a.Gender
FROM Movies m, Actors a
WHERE m.MovieTitle = a.MovieTitle
AND m.Genre = 'Drama'
AND a.Age > 30
AND m.ReleaseYear >= 1990
AND a.Name LIKE '%Robbins%';
```

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

MovieTitle	Genre	ReleaseYear	Director	Name	Age	Gender
The Shawshank Redemption	Drama	1994	Frank Darabont	Tim Robbins	62	Male

Pair 3:

Table 1- employees

SQL File 2* | first - Table

Limit to 10 rows

```
select * from employees
limit 5;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell C

EmployeeID	EmployeeName	Department	Salary
1	John Doe	Sales	50000.00
2	Jane Smith	Marketing	60000.00
3	Michael Johnson	Finance	75000.00
4	Emily Williams	HR	55000.00
5	Robert Brown	IT	80000.00

Table 2- Departments

SQL File 2* | first - Table

Limit to 10 rows

```
select * from departments
limit 5;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: | Fetch rows:

DepartmentID	Department	Location
1	Sales	New York
2	Marketing	Los Angeles
3	Finance	Chicago
4	HR	San Francisco
5	IT	Seattle

SQL Query:

SQL File 2* | first - Table

Limit to 10 rows

```sql
SELECT Employees.EmployeeID, Employees.EmployeeName, Employees.Salary,
Departments.Location, Departments.Department
FROM Employees
JOIN Departments
ON Employees.Department = Departments.Department
WHERE Employees.Department LIKE '%F%'
AND Departments.Location = 'Chicago';
```

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

EmployeeID	EmployeeName	Salary	Location	Department
3	Michael Johnson	75000.00	Chicago	Finance

Pair 4:

Table 1- InfosysEmployees

SQL File 2* | first - Table

Limit to 10 rows

```
select * from InfosysEmployees
limit 5;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: | Fetch

EmployeeID	EmployeeName	Age	Department
1	Alex Johnson	30	HR
2	Samantha Lee	35	Marketing
3	Richard Williams	28	Finance
4	Laura Brown	40	IT
5	Matthew Smith	32	Sales

InfosysEmployees 74

Table 2- CognizantEmployees

SQL File 2* | first - Table

Limit to 10 rows

```
select * from CognizantEmployees
limit 5;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: | Fet

EmployeeID	EmployeeName	Age	Department
1	John Doe	30	HR
2	Jane Smith	35	Marketing
3	Michael Johnson	28	Finance
4	Emily Williams	40	IT
5	Robert Brown	32	Sales

CognizantEmployees 75

SQL Query:

SQL File 2* | first - Table

Limit to 10 rows

```sql
SELECT ce.EmployeeID, ce.EmployeeName AS CognizantEmployee, ie.EmployeeName
AS InfosysEmployee, ce.Department
FROM CognizantEmployees ce
JOIN InfosysEmployees ie ON ce.Department = ie.Department
WHERE ce.EmployeeName LIKE '%J%'
AND ie.EmployeeName LIKE '%L%'
AND ce.Department = 'IT';
```

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

EmployeeID	CognizantEmployee	InfosysEmployee	Department
8	Jennifer Brown	Laura Brown	IT

Pair 5:

Table 1- Customers

SQL File 2* | first - Table

Limit to 10 rows

```sql
select * from customers
limit 5;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: | Fetch rows:

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico
4	Around the Horn	Thomas Hardy	120 Hanover Sq.	London	WA1 1DP	UK
5	Berglunds snabbköp	Christina Berglund	Berguvsvägen 8	Luleå	S-958 22	Sweden

customers 77

Table 2- Orders

SQL File 2* | first - Table

Limit to 10 rows

```sql
select * from orders
limit 5;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

OrderID	CustomerID	EmployeeID	OrderDate	ShipperID	City
1	1	3	2023-11-01	1	Berlin
2	2	4	2023-11-02	2	México D.F.
3	3	3	2023-11-03	1	México D.F.
4	4	4	2023-11-04	2	London
5	5	3	2023-11-05	3	Luleå

orders 78

SQL Query:

SQL File 2* | first - Table

Limit to 10 rows

```sql
SELECT
    c.CustomerName, c.City AS CustomerCity, o.OrderID, o.OrderDate, o.City AS OrderCity
FROM Customers c
JOIN Orders o ON c.CustomerID = o.CustomerID
WHERE c.City = o.City
AND c.Country LIKE '%M%'
AND o.OrderDate > '2023-11-02';
```

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

CustomerName	CustomerCity	OrderID	OrderDate	OrderCity
Antonio Moreno Taquería	México D.F.	3	2023-11-03	México D.F.