



UNIVERSITY OF CHITTAGONG

Department of Computer Science & Engineering

Program: **B.Sc. (Engineering)**

Session: 2022-2023

4th Semester

Tutorial_01

Topic: DataBase Modeling

Course Title: DataBase Systems Lab

Course Code: CSE - 414

Submitted To:

Dr. Rudra Pratap Deb Nath

Associate Professor

Department of Computer Science & Engineering

University of Chittagong

Submitted By:

Nilanjana Das Jui

ID: 23701011

Dept. of Computer Science & Engineering

Date of submission: June 23, 2025

Contents

- Task - 01: Identify Relevant Data2
- Task - 02: Entity - Relationship Diagram 3
- Task - 03: Entity - Relationship Model4
- Task - 04: Full Procedure of DataBase Modeling5
- Task - 05: Procedure Flowchart of Database Modeling 8
- Task - 06: Queries8

Task - 01:

Identify Relevant Data

Based on the dataset and the queries I needed to solve, I considered these data items important:

- Drug: {drug name, drug category}
- Disease: {disease name, disease category}
- Product: {product name, company name}
- Side effects (can be more than one)
- Clinical trial: title, start date, completion date, number of participants, status
- Researcher: {name}
- Institution: {name, address, country}
- Relationships:
 - A drug can treat one or more diseases
 - A drug can have multiple side effects
 - A drug can be tested in many clinical trials
 - A clinical trial is led by one researcher
 - A clinical trial is held in one institution

Task - 02: Entity - Relationship Diagram

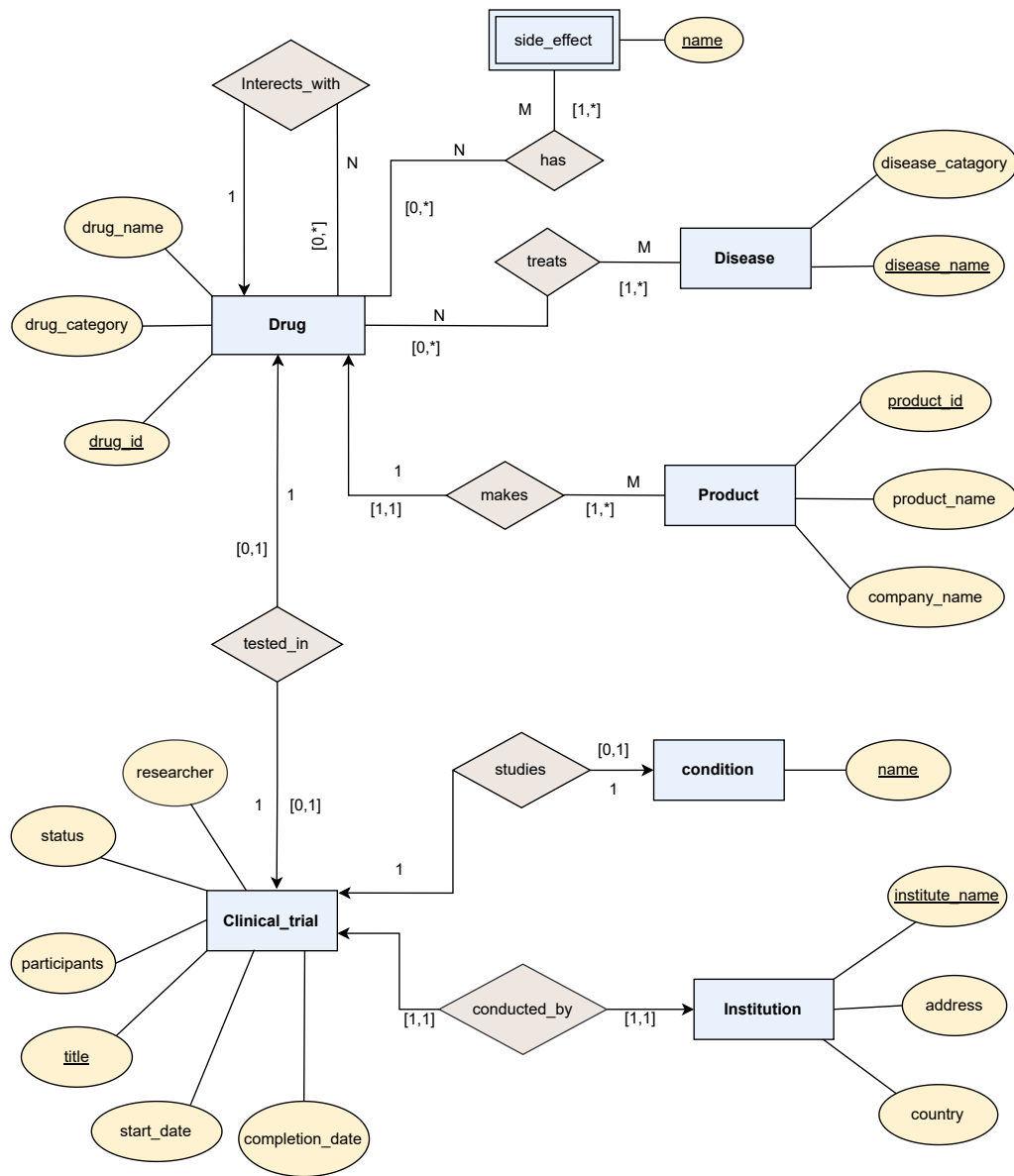


Figure 1: Entity Relationship Diagram

Task - 03:

Entity - Relationship Model

Entity Types

- **Drug**
 $\{\{\underline{\text{drug_id}}, \text{drug_name}, \text{drug_category}\}\}$
- **Disease**
 $\{\{\underline{\text{disease_name}}, \text{disease_category}, \text{drug_id (FK)}\}\}$
- **Product**
 $\{\{\underline{\text{product_id}}, \text{product_name}, \text{company_name}, \text{drug_id (FK)}\}\}$
- **side_effect**
 $\{\{\underline{\text{name}}\}\}$
- **condition**
 $\{\{\underline{\text{name}}\}\}$
- **Institution**
 $\{\{\underline{\text{institute_name}}, \text{address}, \text{country}\}\}$
- **Clinical_trial**
 $\{\{\underline{\text{title}}, \text{researcher}, \text{status}, \text{participants}, \text{start_date}, \text{completion_date}, \text{institute_name (FK)}\}\}$

Relationship Types

- **treats**
 $\{\{\underline{\text{drug_id} \rightarrow \text{Drug}}, \underline{\text{disease_name} \rightarrow \text{Disease}}\}\}$
Drug treats Disease (M:N)
- **has**
 $\{\{\underline{\text{drug_id} \rightarrow \text{Drug}}, \underline{\text{name} \rightarrow \text{side_effect}}\}\}$
Drug has side_effect (1:M or M:N)
- **interacts_with**
 $\{\{\underline{\text{drug_id1} \rightarrow \text{Drug}}, \underline{\text{drug_id2} \rightarrow \text{Drug}}\}\}$
Drug interacts with Drug (recursive 1:N)
- **tested_in**
 $\{\{\underline{\text{drug_id} \rightarrow \text{Drug}}, \underline{\text{title} \rightarrow \text{Clinical_trial}}\}\}$
Drug is tested in Clinical_trial (1:1)
- **studies**
 $\{\{\underline{\text{title} \rightarrow \text{Clinical_trial}}, \underline{\text{name} \rightarrow \text{condition}}\}\}$
Clinical_trial studies condition (1:1)

Task - 04:

Full Procedure of DataBase Modeling

1. Create Entity-Relationship (ER) Model

- Analyzed the dataset.
- Identified:
 - Strong entities: Drug, Disease, Product, ClinicalTrial, Researcher, Institution
 - Weak/multivalued attributes: side effects, interacts_with
 - Relationships: Treats, HasSideEffect, TestedIn, Makes, LedBy, etc.
- Designed ER diagram in draw.io based on normalized structure.
- Discussed total/partial participation, weak vs strong entities, cardinality.

2. Convert Excel to CSV

- Prepared the raw Excel dataset with columns like:
 - Drug, Disease, Side Effects, Interactions, Trial Info, Researcher, Address
- Saved the Excel file as .csv for Python processing.

3. Import Data from CSV Using Python

- Used pandas and oracledb to load CSV.

```
import pandas as pd
import oracledb
```

- Cleaned and mapped data for insertion:
 - drug_id, product_id using enumerate() with unique().
- Inserted rows into DRUGS_FULL master table via SQL loop.

4. Create Master Flat Table: DRUGS_FULL

- Contained all columns (wide schema): drugs, diseases, trials, researchers, side effects, conditions, etc.
- Temporarily acted as the central source for normalization.

5. Create Schema

- Created all normalized entity tables:
 - Drug, Disease, Product, SideEffect, Institution, ClinicalTrial
- Created all relationship tables:
 - Treats, HasSideEffect, TestedIn, Studies, etc.

6. Insert Normalized Data

- Used INSERT INTO ... SELECT DISTINCT ... FROM DRUGS_FULL pattern.
- Handled multivalued attributes via UNION ALL (e.g. side effects, interacts_with).
- Assigned primary key **drug_id** by matching drug_name + drug_category.
- Similarly assigned primary keys and foreign keys for the other tables.

7. Update Entity Tables

- Cleaned and deduplicated data in:
 - Drug: ensured unique drug_id per drug name/category
 - Product: linked to drug_id
- Updated ClinicalTrial:
 - Added **researcher_name** column and **institute_name** column to use as foreign key.

8. Handle Issues (Yes/No Loop)

- Fixed:
 - ORA-00001: duplicate key violations
 - ORA-00942: missing tables.
 - drug_id mismatches between DRUGS_FULL and normalized tables
- Ensured consistent ID generation in Python (no re-generation on rerun)

9. Insert Normalized Data (Again)

- Re-ran INSERT INTO for:
 - Treats, HasSideEffect, TestedIn, Studies etc.
- Ensured joins used TRIM, LOWER, and drug_name + category keys.

10. Update Table Structures

- ALTER TABLE to:
 - Add researcher_name and institute_name to ClinicalTrial
 - Add constraints (PKs, FKs)
- Added foreign key from ClinicalTrial.institute_name → Institution

11. Generate Answers for Query Questions

- Used joins between key tables like Drug, Clinical_trial, and Disease.
- Applied filters, grouping, and aggregate functions to meet specific conditions
- Subqueries and set operations were used to handle overlaps, intersections, and exclusivity.

12. Query Output Correct? (Yes/No Decision)

- If output was wrong:
 - Rebuilt joins (esp. on drug_id).
 - Re-inserted into HasSideEffect, Treats, etc.
- If correct:
 - Moved to next question and stored result.

13. Show Output Tables

- Retrieved and displayed clean query results from normalized schema.
- Verified against expected values from raw Excel file.

Task - 05:

Procedure Flowchart of DataBase Modeling

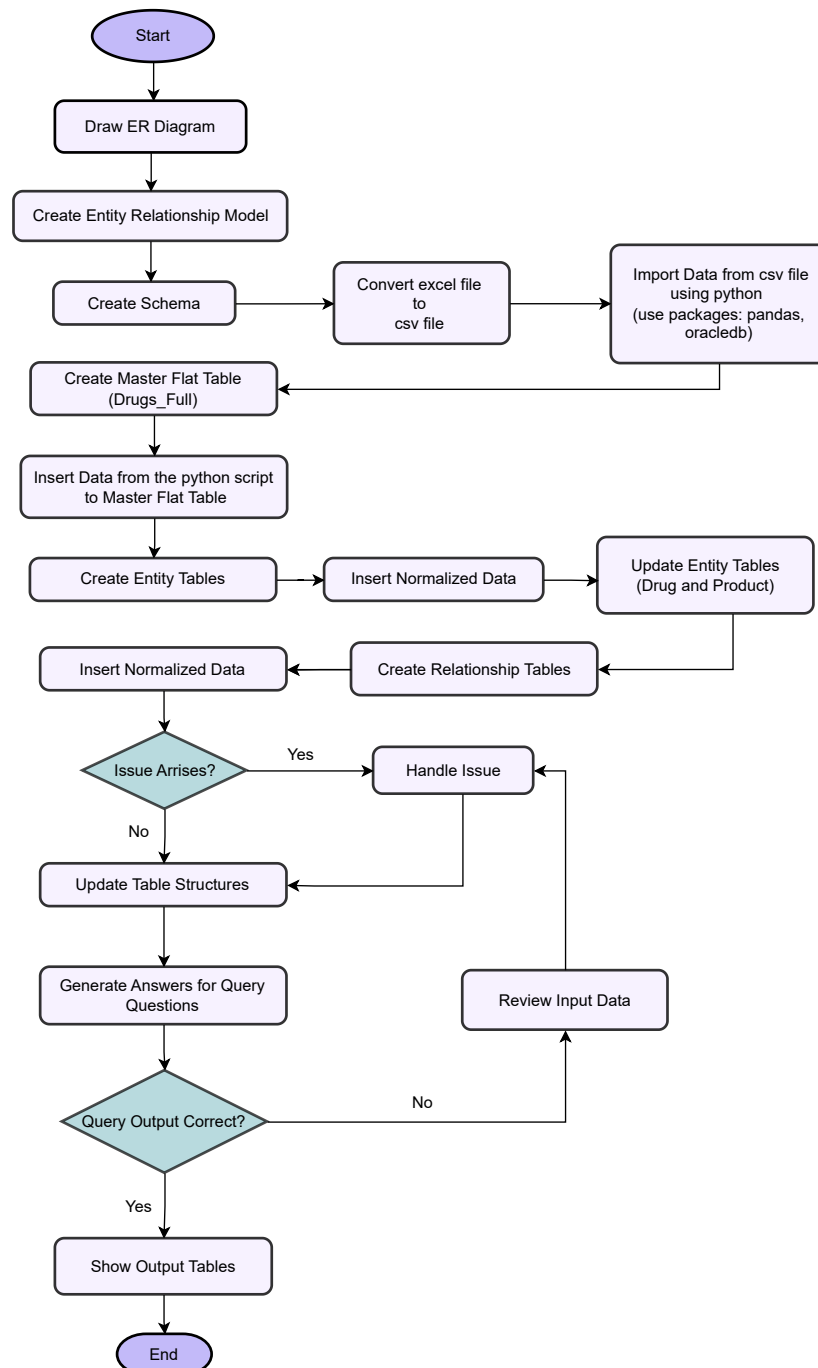
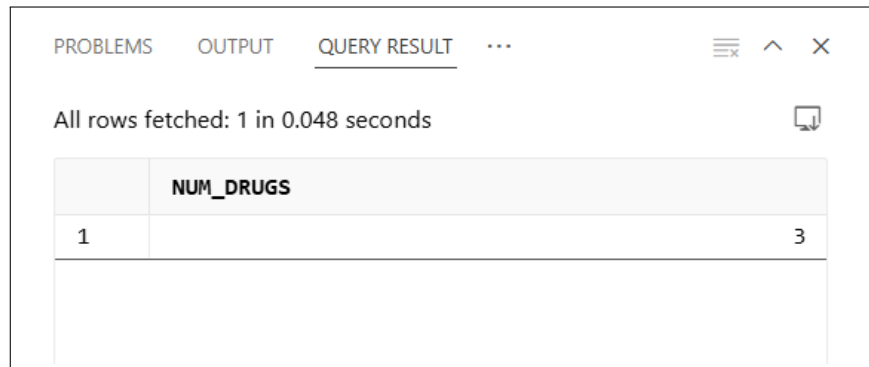


Figure 2: Procedure Flowchart

Task - 06: Queries

a) Find the number of drugs that have nausea as a side effect.

```
SELECT COUNT(DISTINCT drug_id) AS num_drugs
FROM HasSideEffect
WHERE side_effect_name = 'nausea';
```

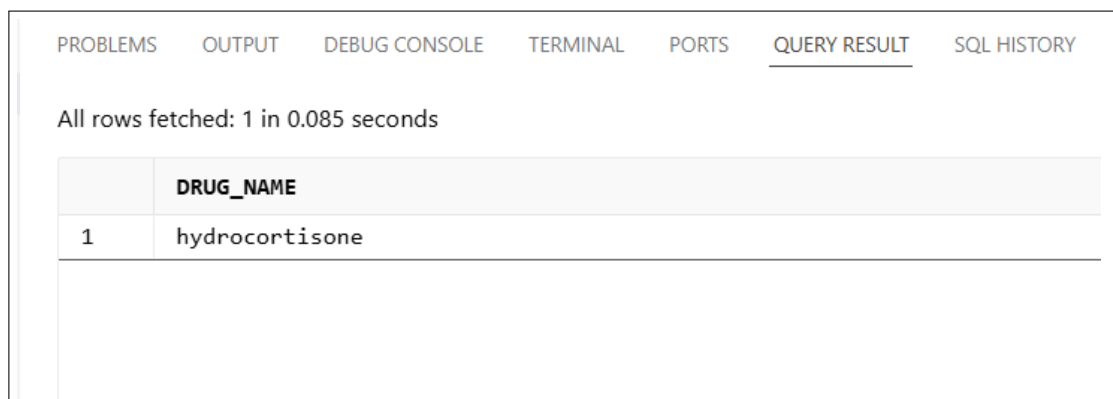


The screenshot shows a SQL query result window with tabs for PROBLEMS, OUTPUT, QUERY RESULT, and a menu icon. The QUERY RESULT tab is active, displaying the text "All rows fetched: 1 in 0.048 seconds" and a copy icon. Below this is a table with one column, NUM_DRUGS, and one row with the value 3.

	NUM_DRUGS
1	3

b) Find the drugs that interact with butabarbital.

```
SELECT DISTINCT drug_name
FROM DRUGS_FULL
WHERE interacts_with1 = 'butabarbital'
OR interacts_with2 = 'butabarbital'
OR interacts_with3 = 'butabarbital';
```

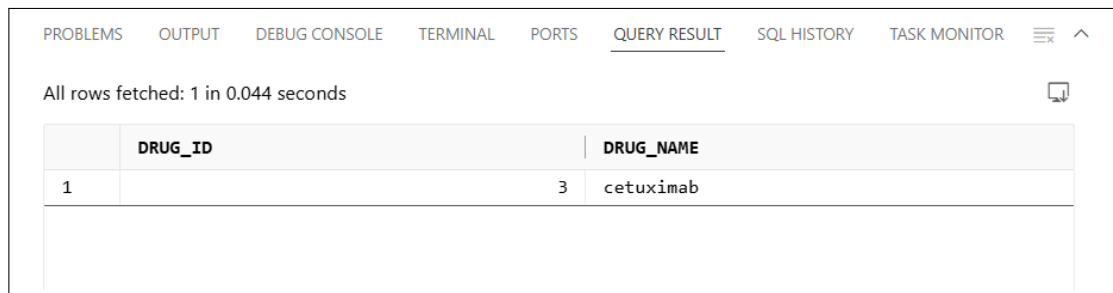


The screenshot shows a SQL query result window with tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, PORTS, QUERY RESULT, and SQL HISTORY. The QUERY RESULT tab is active, displaying the text "All rows fetched: 1 in 0.085 seconds". Below this is a table with one column, DRUG_NAME, and one row with the value hydrocortisone.

	DRUG_NAME
1	hydrocortisone

c) Find the drugs with side effects cough and headache.

```
SELECT d.drug_id, d.drug_name
FROM HasSideEffect hs
JOIN Drug d ON hs.drug_id = d.drug_id
WHERE hs.side_effect_name IN ('cough', 'headache')
GROUP BY d.drug_id, d.drug_name
HAVING COUNT(DISTINCT hs.side_effect_name) = 2;
```

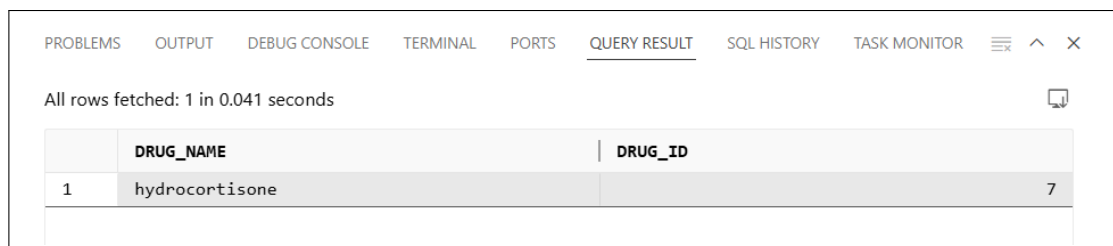


The screenshot shows a database query result window with tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, PORTS, QUERY RESULT, SQL HISTORY, and TASK MONITOR. The 'QUERY RESULT' tab is active, displaying 'All rows fetched: 1 in 0.044 seconds'. The results are shown in a table with two columns: DRUG_ID and DRUG_NAME.

	DRUG_ID	DRUG_NAME
1	3	cetuximab

d) Find the drugs that can be used to treat endocrine diseases.

```
SELECT DISTINCT d.drug_name, d.drug_id
FROM Treats t
JOIN Disease dis ON t.disease_name = dis.disease_name
JOIN Drug d ON t.drug_id = d.drug_id
WHERE LOWER(dis.disease_category) = 'endocrine';
```



The screenshot shows a database query result window with tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, PORTS, QUERY RESULT, SQL HISTORY, and TASK MONITOR. The 'QUERY RESULT' tab is active, displaying 'All rows fetched: 1 in 0.041 seconds'. The results are shown in a table with two columns: DRUG_NAME and DRUG_ID.

	DRUG_NAME	DRUG_ID
1	hydrocortisone	7




e) Find the most common treatment for immunological diseases that have not been used for hematological diseases.


```
SELECT t.drug_id, d.drug_name, COUNT(*) AS usage_count
FROM Treats t
JOIN Disease dis ON t.disease_name = dis.disease_name
JOIN Drug d ON t.drug_id = d.drug_id
WHERE LOWER(dis.disease_category) = 'immunological'
AND NOT EXISTS (
  SELECT 1
  FROM Treats t2
  JOIN Disease d2 ON t2.disease_name = d2.disease_name
```

```

WHERE LOWER(d2.disease_category) = 'hematological'
AND t2.drug_id = t.drug_id
)
GROUP BY t.drug_id, d.drug_name
ORDER BY usage_count DESC;

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RESULT SQL HISTORY TASK MONITOR   

All rows fetched: 2 in 0.083 seconds 

	DRUG_ID	DRUG_NAME	USAGE_COUNT
1	4	denileukin diftitox	5
2	7	hydrocortisone	2

f) Find the diseases that can be treated with hydrocortisone but not with etanercept.

```

SELECT DISTINCT t1.disease_name
FROM Treats t1
JOIN Drug d1 ON t1.drug_id = d1.drug_id
WHERE LOWER(d1.drug_name) = 'hydrocortisone'
AND NOT EXISTS (
    SELECT 1
    FROM Treats t2
    JOIN Drug d2 ON t2.drug_id = d2.drug_id
    WHERE LOWER(d2.drug_name) = 'etanercept'
    AND t2.disease_name = t1.disease_name
);

```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

QUERY RESULT

SQL HISTORY

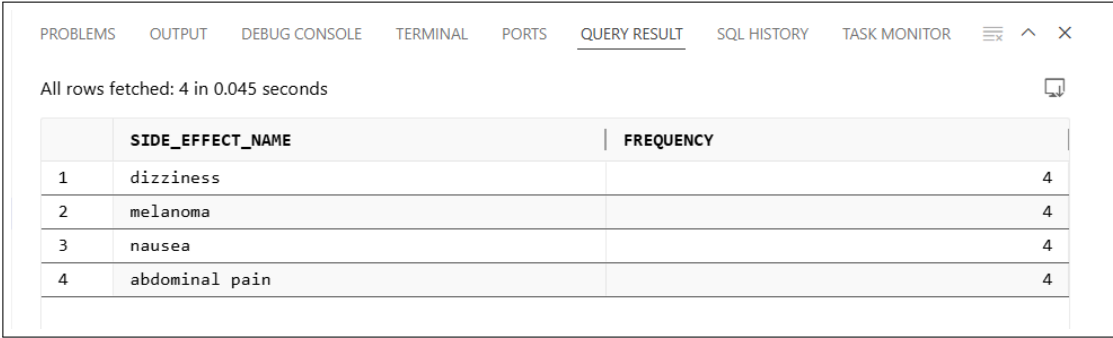
TASK MONITOR

All rows fetched: 7 in 0.044 seconds

	DISEASE_NAME
1	Atherosclerosis, susceptibility to
2	Malaria, resistance to, 248310
3	Obesity, adrenal insufficiency, and red hair
4	Atherosclerosis
5	Cortisol resistance
6	Obesity
7	Asthma, diminished response to antileukotriene treatment in, 600807

g) Find the top-10 side effects that drugs used to treat asthma related diseases have.

```
SELECT hs.side_effect_name, COUNT(*) AS frequency
FROM Treats t
JOIN Disease d ON t.disease_name = d.disease_name
JOIN HasSideEffect hs ON t.drug_id = hs.drug_id
WHERE LOWER(d.disease_name) LIKE '%asthma%'
GROUP BY hs.side_effect_name
ORDER BY frequency DESC;
```

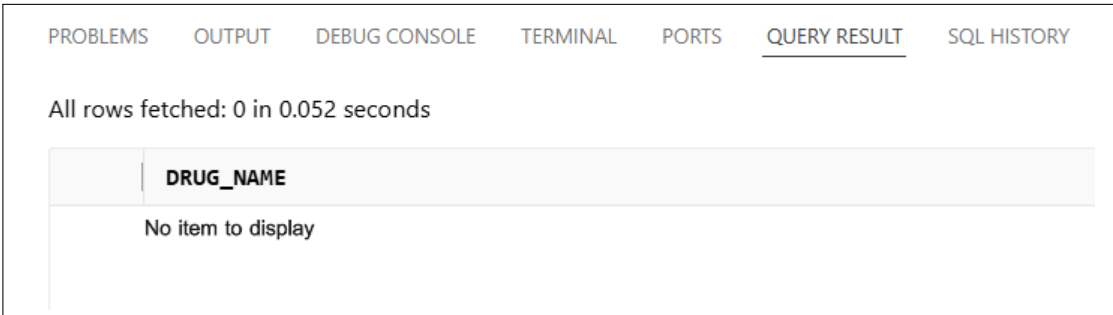


The screenshot shows a database query result window with tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, PORTS, QUERY RESULT, SQL HISTORY, and TASK MONITOR. The 'QUERY RESULT' tab is active, displaying the text 'All rows fetched: 4 in 0.045 seconds'. Below this is a table with two columns: 'SIDE_EFFECT_NAME' and 'FREQUENCY'. The table contains four rows of data.

	SIDE_EFFECT_NAME	FREQUENCY
1	dizziness	4
2	melanoma	4
3	nausea	4
4	abdominal pain	4

h) Find the drugs that have been studied in more than three clinical trials with more than 30 participants.

```
SELECT d.drug_name
FROM TestedIn ti
JOIN ClinicalTrial ct ON ti.clinical_trial_title = ct.clinical_trial_title
JOIN Drug d ON ti.drug_id = d.drug_id
WHERE ct.clinical_trial_participants > 30
GROUP BY d.drug_id, d.drug_name
HAVING COUNT(DISTINCT ti.clinical_trial_title) > 3;
```



The screenshot shows a database query result window with tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, PORTS, QUERY RESULT, and SQL HISTORY. The 'QUERY RESULT' tab is active, displaying the text 'All rows fetched: 0 in 0.052 seconds'. Below this is a table with one column: 'DRUG_NAME'. The table is empty, and the text 'No item to display' is shown.

DRUG_NAME

i) Find the largest number of clinical trials and the drugs they have studied that have been active in the same period of time.

```

WITH TrialActivity AS (
  SELECT
    TRUNC(clinical_trial_start_date + LEVEL - 1) AS active_day,
    clinical_trial_title
  FROM ClinicalTrial
    CONNECT BY LEVEL <= clinical_trial_completion_date - clinical_trial_start_date
    AND PRIOR clinical_trial_title = clinical_trial_title
    AND PRIOR DBMS_RANDOM.VALUE IS NOT NULL
),
PeakDate AS (
  SELECT active_day, COUNT(*) AS trial_count
  FROM TrialActivity
  GROUP BY active_day
  ORDER BY trial_count DESC
  FETCH FIRST 1 ROW ONLY
),
PeakTrials AS (
  SELECT ta.clinical_trial_title
  FROM TrialActivity ta
  JOIN PeakDate pd ON ta.active_day = pd.active_day
),
DrugsInPeakTrials AS (
  SELECT DISTINCT d.drug_name, pt.clinical_trial_title
  FROM PeakTrials pt
  JOIN TestedIn ti ON pt.clinical_trial_title = ti.clinical_trial_title
  JOIN Drug d ON d.drug_id = ti.drug_id
)
SELECT * FROM DrugsInPeakTrials;

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RESULT SQL HISTORY TASK MONITOR		
All rows fetched: 3 in 0.643 seconds		
	DRUG_NAME	CLINICAL_TRIAL_TITLE
1	etanercept	Can We Miss Pigmented Lesions in Psoriasis Patients?
2	cetuximab	Immune Response on Neoadjuvant Therapy in Non-small-cell Lung Cancer (NSCLC)
3	hydrocortisone	Low Doses Corticosteroids as Adjuvant Therapy for the Treatment of Severe H1N1 Flu

j) Find the main researchers that have conducted clinical trials that study drugs that can be used to treat both respiratory and cardiovascular diseases

```
WITH DualCategoryDrugs AS (  
  SELECT t.drug_id  
  FROM Treats t  
  JOIN Disease d ON t.disease_name = d.disease_name  
  WHERE LOWER(d.disease_category) IN ('respiratory', 'cardiovascular')  
  GROUP BY t.drug_id  
  HAVING COUNT(DISTINCT LOWER(d.disease_category)) = 2  
)  
,  
RelevantTrials AS (  
  SELECT DISTINCT ti.clinical_trial_title  
  FROM TestedIn ti  
  JOIN DualCategoryDrugs dc ON ti.drug_id = dc.drug_id  
)  
SELECT DISTINCT ct.researcher_name  
FROM ClinicalTrial ct  
JOIN RelevantTrials rt ON ct.clinical_trial_title = rt.clinical_trial_title  
WHERE ct.researcher_name IS NOT NULL;
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

QUERY RESULT

...

☰


All rows fetched: 2 in 0.051 seconds

📄

	RESEARCHER_NAME
1	Boni Elewski, MD
2	Djillali Annane


k) Find up to three main researchers that have conducted the larger number of clinical trials that study drugs that can be used to treat both respiratory and cardiovascular diseases.

```
WITH DualCategoryDrugs AS (
  SELECT t.drug_id
  FROM Treats t
  JOIN Disease d ON t.disease_name = d.disease_name
  WHERE LOWER(d.disease_category) IN ('respiratory', 'cardiovascular')
  GROUP BY t.drug_id
  HAVING COUNT(DISTINCT LOWER(d.disease_category)) = 2
),
RelevantTrials AS (
  SELECT DISTINCT ti.clinical_trial_title
  FROM TestedIn ti
  JOIN DualCategoryDrugs dc ON ti.drug_id = dc.drug_id
),
ResearcherCounts AS (
  SELECT ct.researcher_name, COUNT(*) AS trial_count
  FROM ClinicalTrial ct
  JOIN RelevantTrials rt ON ct.clinical_trial_title = rt.clinical_trial_title
  WHERE ct.researcher_name IS NOT NULL
  GROUP BY ct.researcher_name
)
SELECT *
FROM ResearcherCounts
ORDER BY trial_count DESC;
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS <u>QUERY RESULT</u> SQL HISTORY ... ≡ ^		
All rows fetched: 2 in 0.043 seconds 		
	RESEARCHER_NAME	TRIAL_COUNT
1	Djillali Annane	1
2	Boni Elewski, MD	1

1) Find the categories of drugs that have been only studied in clinical trials based in United States.

```
SELECT d.drug_category
FROM Drug d
JOIN TestedIn ti ON d.drug_id = ti.drug_id
JOIN ClinicalTrial ct ON ti.clinical_trial_title = ct.clinical_trial_title
JOIN Institution i ON ct.institute_name = i.institute_name
GROUP BY d.drug_category
HAVING COUNT(*) = SUM(
  CASE
    WHEN LOWER(i.country) = 'united states' THEN 1
    ELSE 0
  END
);
```

PROBLEMS	OUTPUT	DEBUG CONSOLE	TERMINAL	PORTS	QUERY RESULT	SQL HISTORY	TASK MONITOR
All rows fetched: 2 in 0.041 seconds							
	DRUG_CATEGORY						
1	Antirheumatic agents						
2	Immunomodulatory agents						