ة

جامعة طيبـــة
TAIBAH UNIVERSITY

CS372 – Database Systems

Sentific content management system

Student name	Student ID
Rand Hamad Alharbi	4455847
Hind Ammar Alharbi	4454429
Shatha Ali Alharbi	4453086
Wedad Khaled Almuhammady	4452851

1. Introduction:

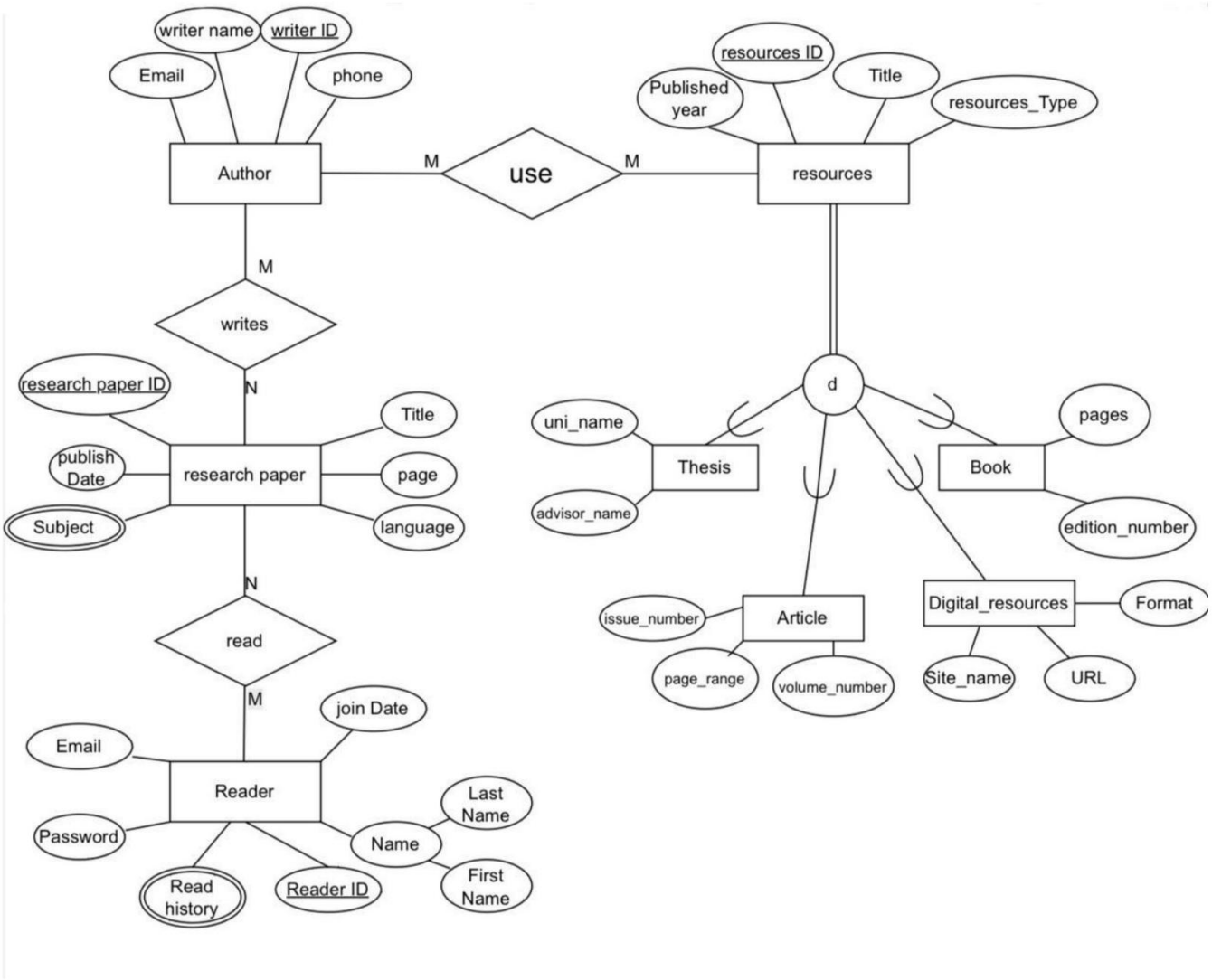
In academic and research settings, the ability to efficiently organize, store, and retrieve scholarly materials is essential to supporting learning, teaching, and scholarly work. Universities, libraries, and research groups often deal with a wide range of resources—including articles, research papers, books, dissertations, and digital documents—which can quickly become difficult to manage without a structured system. When this information is stored using disorganized files, scattered folders, or manual records, users face challenges such as duplicate data, missing details, inconsistent updates, and limited search capabilities.

This project addresses these issues by designing a comprehensive database system for managing scholarly content, providing a reliable framework for managing academic resources. The system supports multiple key entities, including authors, readers, and various types of scholarly materials, allowing for the accurate documentation of each resource with attributes such as publication year, title, edition details, URLs, figures, and page numbers. It also manages the relationships between authors and their publications, enabling users to understand how each research product is produced, linked to, and utilized.

By developing a detailed Enhanced Entity Relationship (EER) model, the project clearly defines entities, attributes, cardinalities, and specialization structures. This conceptual model is then translated into a relational diagram, ensuring data integrity, consistency, and normalization. Finally, SQL DDL and DML commands are used to implement the database, populate it with sample data, and execute basic queries such as SELECT, INSERT, UPDATE, and JOIN.

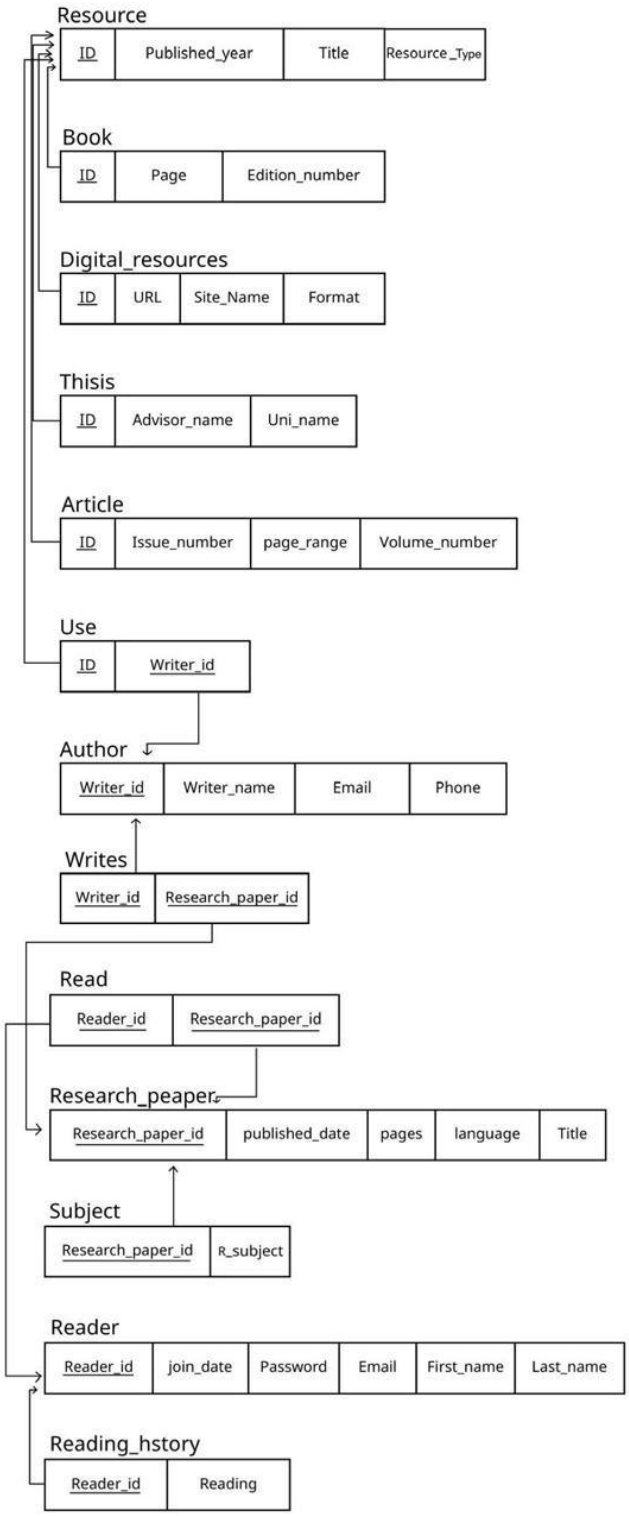
Overall, the project provides a structured and scalable solution for managing academic content, offering improved data accuracy, accessibility, and enhanced support for users who rely on scientific resources for study, research, and knowledge production.

2. Conceptual Model Design



3. Relational Model Design

3.1.



3.2 . steps for converting EER model into Relational schema:

Algorithm:

Step 1: Mapping of Regular Entity Types
For each regular (strong) entity type in the ER schema, a relation is created that includes all simple attributes of the entity. One of the key attributes is selected as the primary key.
Author(Writer_id PK, Writer_name, Email, Phone)
Reader(Reader_id PK, First_name, Last_name, Email, Password, Join_date)
Research_paper(Research_paper_id PK, Title, Published_date, Pages, Language)
Resource(ID PK, Title, Published_year, Resource_type)

Step 2: Mapping of Weak Entity Types
There are no weak entity types in the given ER/EER diagram. Therefore, this step is not applicable.

Step 3: Mapping of Binary 1:1 Relationship Types
There are no binary 1:1 relationship types in the given ER/EER diagram. Therefore, this step is not applicable.

Step 4: Mapping of Binary 1:N Relationship Types
There are no binary 1:N relationship types in the given ER/EER diagram. Therefore, this step is not applicable.

Step 5: Mapping of Binary M:N Relationship Types
For each binary M:N relationship type, a new relation is created. The primary key of this relation is a combination of the primary keys of the participating entities.
Use(Writer_id PK/FK → Author.Writer_id, ID PK/FK → Resource.ID)
Writes(Writer_id PK/FK → Author.Writer_id, Research_paper_id PK/FK → Research_paper.Research_paper_id)
Read(Reader_id PK/FK → Reader.Reader_id, Research_paper_id PK/FK → Research_paper.Research_paper_id)

Step 6: Mapping of Multivalued Attributes
For each multivalued attribute, a separate relation is created.
The relation includes the multivalued attribute and the primary key of the owning entity.
The combination of these attributes forms the primary key.
The multivalued attributes identified in the ER/EER diagram are:
Subject(
Research_paper_id PK/FK → Research_paper.Research_paper_id,
R_subject PK)

Reading_history(
Reader_id PK/FK → Reader.Reader_id,
Reading PK)

Step 7: Mapping of N-ary Relationship Types
There are no n-ary (n > 2) relationship types in the given ER/EER diagram. Therefore, this step is not applicable.

Step 8: Mapping of Specialization / Generalization
The Resource entity is specialized into Book, Article, Thesis, and Digital_resources.
Option 8A (Multiple relations – superclass and subclasses) is applied.
Book(ID PK/FK, Pages, Edition_number)
Article(ID PK/FK, Issue_number, Page_range, Volume_number)
Thesis(ID PK/FK, Advisor_name, Uni_name)
Digital_resources(ID PK/FK, URL, Site_name, Format)

Step 9: Mapping of Union Types (Categories)
There are no union types (categories) in the given ER/EER diagram. Therefore, this step is not applicable

---

4. Database Implementation and SQL Queries

```
CREATE TABLE Reader (FName VARCHAR2(20) NOT NULL , LastName VARCHAR2(20) NOT NULL,
ReaderID NUMBER(10)PRIMARY KEY,

Password VARCHAR2(30)NOT NULL,Email VARCHAR2 (20)NOT NULL,JoinDate DATE);


CREATE TABLE ResearchPaper(ResearchPaperID NUMBER(10) PRIMARY KEY ,

Title VARCHAR2(20)NOT NULL,Pages NUMBER(5), Language VARCHAR2(20)

,PablishDate DATE);


CREATE TABLE Subject( ResearchPaperID NUMBER(10) PRIMARY KEY,

SubjectName VARCHAR2(20)NOT NULL,

FOREIGN KEY (ResearchPaperID) REFERENCES  ResearchPaper(ResearchPaperID));
```

```
CREATE TABLE Read_History(
Reading NUMBER(10) PRIMARY KEY ,
ReaderID  NUMBER(10) ,
FOREIGN KEY (ReaderID) REFERENCES Reader(ReaderID)
);

 CREATE TABLE ReadRelation (
  ReaderID NUMBER(10),
  ResearchPaperID NUMBER(10),
  PRIMARY KEY (ReaderID, ResearchPaperID),
 FOREIGN KEY (ReaderID) REFERENCES Reader(ReaderID),
  FOREIGN KEY (ResearchPaperID) REFERENCES ResearchPaper(ResearchPaperID)
);
INSERT INTO ResearchPaper VALUES(101, 'AI', 25, 'English', DATE '2023-10-10');
INSERT INTO ResearchPaper VALUES(102, 'Math', 40, 'English', DATE '2022-11-15');
INSERT INTO ResearchPaper VALUES(103, 'Physics', 30, 'Arabic', DATE '2021-12-20');
INSERT INTO ResearchPaper VALUES(104, 'Biology', 35, 'English', DATE '2020-09-05');
INSERT INTO ResearchPaper VALUES(105, 'CS', 45, 'Arabic', DATE '2024-01-25');

INSERT INTO Reader VALUES(1, 'Sara', 'Ali', 'pass123', 'sara@example.com', DATE '2024-01-10');
INSERT INTO Reader VALUES(2, 'Lama', 'Hassan', 'lama2024', 'lama@example.com', DATE '2024-02-05');
INSERT INTO Reader VALUES(3, 'Nora', 'Saleh', 'nora999', 'nora@example.com', DATE '2024-03-01');
INSERT INTO Reader VALUES(4, 'Maha', 'Omar', 'maha777', 'maha@example.com', DATE '2024-04-11');
INSERT INTO Reader VALUES(5, 'Rana', 'Fahad', 'rana555', 'rana@example.com', DATE '2024-05-20');

INSERT INTO Subject VALUES (101, 'AI');
INSERT INTO Subject VALUES (102, 'Math');
INSERT INTO Subject VALUES (103, 'Physics');
INSERT INTO Subject VALUES (104, 'Biology');
```

```
INSERT INTO Subject VALUES (105, 'CS');

INSERT INTO Read_History VALUES (1001, 1);
INSERT INTO Read_History VALUES (1002, 2);
INSERT INTO Read_History VALUES (1003, 3);
INSERT INTO Read_History VALUES (1004, 4);
INSERT INTO Read_History VALUES (1005, 5);

INSERT INTO ReadRelation VALUES (1, 101);
INSERT INTO ReadRelation VALUES (1, 102);
INSERT INTO ReadRelation VALUES (2, 103);
INSERT INTO ReadRelation VALUES (3, 101);
INSERT INTO ReadRelation VALUES (5, 105);

CREATE TABLE Writes (
   Writer_id       NUMBER,
   ResearchPaperID  NUMBER(10),

   CONSTRAINT writes_pk
      PRIMARY KEY (Writer_id, ResearchPaperID),

   CONSTRAINT writes_fk_author
      FOREIGN KEY (Writer_id)
      REFERENCES AUTHOR(WRITER_ID)
      ON DELETE CASCADE,

   CONSTRAINT writes_fk_paper
      FOREIGN KEY (ResearchPaperID)
      REFERENCES ResearchPaper(ResearchPaperID)
```

```
    ON DELETE CASCADE
);
CREATE TABLE AUTHOR (
  WRITER_ID     NUMBER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,
  WRITER_NAME   VARCHAR2(100) NOT NULL,
  EMAIL         VARCHAR2(120),
  PHONE         VARCHAR2(30)
);
INSERT INTO AUTHOR (WRITER_NAME, EMAIL, PHONE)
VALUES ('Sara Writer', 'sara.writer@example.com', '0501111111');

INSERT INTO AUTHOR (WRITER_NAME, EMAIL, PHONE)
VALUES ('Lama Alharbi', 'lama.writer@example.com', '0502222222');

INSERT INTO AUTHOR (WRITER_NAME, EMAIL, PHONE)
VALUES ('Nora Saleh', 'nora.writer@example.com', '0503333333');

INSERT INTO AUTHOR (WRITER_NAME, EMAIL, PHONE)
VALUES ('Maha Omar', 'maha.writer@example.com', '0504444444');

INSERT INTO AUTHOR (WRITER_NAME, EMAIL, PHONE)
VALUES ('Rana Fahad', 'rana.writer@example.com', '0505555555');

INSERT INTO Writes (Writer_id, ResearchPaperID)VALUES (1, 101);
INSERT INTO Writes (Writer_id, ResearchPaperID)VALUES (2, 102);
INSERT INTO Writes (Writer_id, ResearchPaperID)VALUES (3, 103);
INSERT INTO Writes (Writer_id, ResearchPaperID)VALUES (4, 104);
INSERT INTO Writes (Writer_id, ResearchPaperID)VALUES (5, 105);
```

```
CREATE TABLE Resources (
    resource_id NUMBER(10) CONSTRAINT resources_pk PRIMARY KEY,
    title VARCHAR2(200)CONSTRAINT title_NotNull NOT NULL,
    resource_type VARCHAR2(20)CONSTRAINT resource_type_Notnull NOT NULL,
    published_year NUMBER(4),


    CONSTRAINT chack_resource_type CHECK (resource_type IN ('ARTICLE', 'BOOK', 'THESIS',
'DIGITAL'))
);

INSERT INTO Resources (resource_id, title, resource_type, published_year )
VALUES (1,'Attention Is All You Need','ARTICLE', 2017 );
INSERT INTO Resources (resource_id, title,  resource_type, published_year)
VALUES (2,'The Thousand Brains Theory of Intelligence',  'ARTICLE', 2021);
INSERT INTO Resources (resource_id, title, resource_type, published_year )
VALUES (3,'Symbolic Regression via Genetic Programming', 'ARTICLE', 1992);
INSERT INTO Resources (resource_id, title, resource_type, published_year)
VALUES (4,'Self-Improving Large Language Models via Reflection and RL', 'ARTICLE', 2024);
INSERT INTO Resources (resource_id,title, resource_type, published_year)
VALUES (5,'Neuromorphic Computing and Spiking Neural Networks', 'ARTICLE', 2015);

CREATE TABLE Article (
    resource_id NUMBER(10) CONSTRAINT pk_article PRIMARY KEY,
    page_range VARCHAR2(30),
    issue_number NUMBER(10),
    volume_number NUMBER(10),

    CONSTRAINT fk_article_resource FOREIGN KEY (resource_id) REFERENCES Resources(resource_id)
```

```
);

INSERT INTO Article (resource_id, page_range, issue_number, volume_number)
VALUES (1, '1-11', 12, 45);
INSERT INTO Article (resource_id, page_range, issue_number, volume_number)
VALUES (2, '50-68', 3, 7);
INSERT INTO Article (resource_id, page_range, issue_number, volume_number)
VALUES (3, '141-157', 2, 1);
INSERT INTO Article (resource_id, page_range, issue_number, volume_number)
VALUES (4, '1-25', 5, 2);
INSERT INTO Article (resource_id, page_range, issue_number, volume_number)
VALUES (5, '85-104', 10, 22);

CREATE TABLE Author_Use_Resources (
resource_id NUMBER(20),
WRITER_ID NUMBER(20),

CONSTRAINT pk_author_resources
      PRIMARY KEY (writer_id, resource_id),

   CONSTRAINT fk_ar_author
      FOREIGN KEY (writer_id)
      REFERENCES AUTHOR( WRITER_ID),

   CONSTRAINT fk_ar_resources
      FOREIGN KEY (resource_id)
      REFERENCES ResearchPaper(ResearchPaperID)
);
```

```
INSERT INTO Author_Use_Resources (writer_id, resource_id)
VALUES (1, 101);
INSERT INTO Author_Use_Resources (writer_id, resource_id)
VALUES (2, 102);
INSERT INTO Author_Use_Resources (writer_id, resource_id)
VALUES (3, 103);
INSERT INTO Author_Use_Resources (writer_id, resource_id)
VALUES (4, 104);
INSERT INTO Author_Use_Resources (writer_id, resource_id)
VALUES (5, 105);


CREATE TABLE Thesis (
   resource_id NUMBER(10) CONSTRAINT pk_thesis PRIMARY KEY,
   university_name VARCHAR2(150),    advisor_name VARCHAR2(150),
   CONSTRAINT fk_thesis_resource FOREIGN KEY (resource_id)
      REFERENCES ResearchPaper(ResearchPaperID)
);


INSERT INTO Resources (resource_id, title, resource_type, published_year )
VALUES (11, 'Optimization of Deep Neural Networks Using Hybrid Genetic Algorithms','THESIS' ,2023);
INSERT INTO Resources (resource_id, title, resource_type, published_year )
VALUES (12, 'Impact of Social Media on Academic Performance Among University Students','THESIS' ,
2022);
INSERT INTO Resources (resource_id, title, resource_type, published_year )
VALUES (13, 'Design of Smart Waste Management System Using IoT','THESIS' , 2021);
INSERT INTO Resources (resource_id, title, resource_type, published_year )
VALUES (14, 'Machine Learning for Medical Image Segmentation','THESIS' , 2024);
INSERT INTO Resources (resource_id, title, resource_type, published_year )
```

```
VALUES (15, 'Blockchain-Based Identity Verification System','THESIS' , 2023);

INSERT INTO Thesis (resource_id, university_name, advisor_name)
VALUES (101, 'King Saud University', 'Dr. Noura Al-Qahtani');
INSERT INTO Thesis (resource_id, university_name, advisor_name)
VALUES (102, 'Taibah University', 'Dr. Maha Al-Zahrani');
INSERT INTO Thesis (resource_id, university_name, advisor_name)
VALUES (103, 'King Fahd University of Petroleum and Minerals', 'Dr. Abdulrahman Al-Ghamdi');
INSERT INTO Thesis (resource_id, university_name, advisor_name)
VALUES (104, 'King Abdulaziz University', 'Dr. Hanan Al-Amri');
INSERT INTO Thesis (resource_id, university_name, advisor_name)
VALUES (105, 'Prince Sultan University', 'Dr. Khalid Al-Tamimi');

CREATE TABLE Book (
   resource_id NUMBER(10),
   edition_number NUMBER,
   pages NUMBER,
   CONSTRAINT pk_book PRIMARY KEY (resource_id, edition_number),
   CONSTRAINT fk_book_resource FOREIGN KEY (resource_id)
      REFERENCES ResearchPaper(ResearchPaperID)
);

INSERT INTO Resources (resource_id, title, resource_type, published_year )
VALUES (101, 'Deep Learning','BOOK' ,2016);

INSERT INTO Resources (resource_id, title, resource_type, published_year )
VALUES (7, 'Atomic Habits','BOOK', 2018);

INSERT INTO Resources (resource_id, title, resource_type, published_year )
```

```
VALUES (8, 'Clean Code: A Handbook of Agile Software Craftsmanship', 'BOOK',2008);

INSERT INTO Resources (resource_id, title, resource_type, published_year )
VALUES (9, 'The Pragmatic Programmer', 'BOOK',1999);

INSERT INTO Resources (resource_id, title, resource_type, published_year )
VALUES (10, 'Thinking, Fast and Slow','BOOK', 2011);

INSERT INTO Book (resource_id, edition_number, pages)
VALUES (101, 1, 775);

INSERT INTO Book (resource_id, edition_number, pages)
VALUES (102, 1, 320);

INSERT INTO Book (resource_id, edition_number, pages)
VALUES (103, 1, 464);

INSERT INTO Book (resource_id, edition_number, pages)
VALUES (104, 2, 352);

INSERT INTO Book (resource_id, edition_number, pages)
VALUES (105, 1, 499);

SELECT a.WRITER_ID, a.WRITER_NAME, rp.ResearchPaperID, rp.Title AS Title_Research_Paper
FROM AUTHOR a INNER JOIN Writes w ON a.WRITER_ID = w.Writer_id
INNER JOIN ResearchPaper rp ON w.ResearchPaperID = rp.ResearchPaperID;

    SELECT a.WRITER_ID, a.WRITER_NAME,w.ResearchPaperID
FROM AUTHOR a LEFT JOIN Writes w ON a.WRITER_ID = w.Writer_id;
```

```
ELECT resource_id, title, published_year
FROM Resources
WHERE published_year > 2010;

SELECT resource_type, COUNT(*) AS num_resources
FROM Resources
GROUP BY resource_type
HAVING COUNT(*) > 2;

SELECT
   a.WRITER_ID,
   a.WRITER_NAME,
   COUNT(w.ResearchPaperID) AS Total_Papers
FROM AUTHOR a
LEFT JOIN Writes w
   ON a.WRITER_ID = w.Writer_id
GROUP BY a.WRITER_ID, a.WRITER_NAME
ORDER BY Total_Papers DESC;

SELECT
   resource_id,
   SUM(pages) AS Total_Pages_All_Editions
FROM Book
GROUP BY resource_id
ORDER BY Total_Pages_All_Editions DESC;
```

4. Output Queries

- Joins between multiple tables (INNER JOIN ,LEFT JOIN)

```
INSERT INTO Read_History VALUES (1005, 5);

INSERT INTO ReadRelation VALUES (1, 101);
INSERT INTO ReadRelation VALUES (1, 102);
INSERT INTO ReadRelation VALUES (2, 103);
INSERT INTO ReadRelation VALUES (3, 101);
INSERT INTO ReadRelation VALUES (5, 105);

SELECT a.WRITER_ID, a.WRITER_NAME, rp.ResearchPaperID, rp.Title AS Title_Research_Paper
FROM AUTHOR a INNER JOIN Writes w ON a.WRITER_ID = w.Writer_id
INNER JOIN ResearchPaper rp ON w.ResearchPaperID = rp.ResearchPaperID;

SELECT R.resource_id, R.title, R.published_year,
       T.university_name, T.advisor_name
FROM Resources R, Thesis T
 where R.resource_id = T.resource_id
ORDER BY R.published_year DESC;

SELECT R.resource_id, R.title, B.edition_number, B.pages
```

Query Result

SQL | All Rows Fetched: 5 in 0.025 seconds

	WRITER_ID	WRITER_NAME	RESEARCHPAPERID	TITLE_RESEARCH_PAPER
1	1	Sara Writer	101	AI
2	2	Lama Alharbi	102	Math
3	3	Nora Saleh	103	Physics
4	4	Maha Omar	104	Biology
5	5	Rana Fahad	105	CS

```
INSERT INTO ReadRelation VALUES (1, 101);
INSERT INTO ReadRelation VALUES (1, 102);
INSERT INTO ReadRelation VALUES (2, 103);
INSERT INTO ReadRelation VALUES (3, 101);
INSERT INTO ReadRelation VALUES (5, 105);

SELECT a.WRITER_ID, a.WRITER_NAME, rp.ResearchPaperID, rp.Title AS Title_Research_Paper
FROM AUTHOR a INNER JOIN Writes w ON a.WRITER_ID = w.Writer_id
INNER JOIN ResearchPaper rp ON w.ResearchPaperID = rp.ResearchPaperID;

SELECT a.WRITER_ID, a.WRITER_NAME,w.ResearchPaperID
FROM AUTHOR a LEFT JOIN Writes w ON a.WRITER_ID = w.Writer_id;

SELECT R.resource_id, R.title, R.published_year,
       T.university_name, T.advisor_name
FROM Resources R, Thesis T
 where R.resource_id = T.resource_id
```

Query Result | SQL | All Rows Fetched: 5 in 0.006 seconds

	WRITER_ID	WRITER_NAME	RESEARCHPAPERID
1	1	Sara Writer	101
2	2	Lama Alharbi	102
3	3	Nora Saleh	103
4	4	Maha Omar	104
5	5	Rana Fahad	105

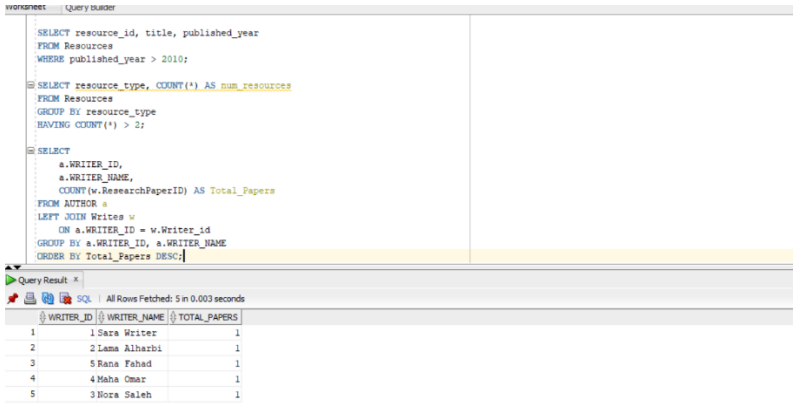
-Basic retrievals using SELECT-FROM-WHERE

```
       T.university_name, T.advisor_name
FROM Resources R, Thesis T
 where R.resource_id = T.resource_id
ORDER BY R.published_year DESC;

SELECT R.resource_id, R.title, B.edition_number, B.pages
FROM Resources R
JOIN Book B ON R.resource_id = B.resource_id
ORDER BY B.pages DESC;

SELECT resource_id, title, published_year
FROM Resources
WHERE published_year > 2010;

SELECT resource_type, COUNT(*) AS num_resources
FROM Resources
GROUP BY resource_type
HAVING COUNT(*) > 2;
```

Query Result | SQL | All Rows Fetched: 4 in 0.002 seconds

	RESOURCE_ID	TITLE	PUBLISHED_YEAR
1	1	Attention Is All You Need	2017
2	2	The Thousand Brains Theory of Intelligence	2021
3	4	Self-Improving Large Language Models via Reflection and RL	2024
4	5	Neuromorphic Computing and Spiking Neural Networks	2015

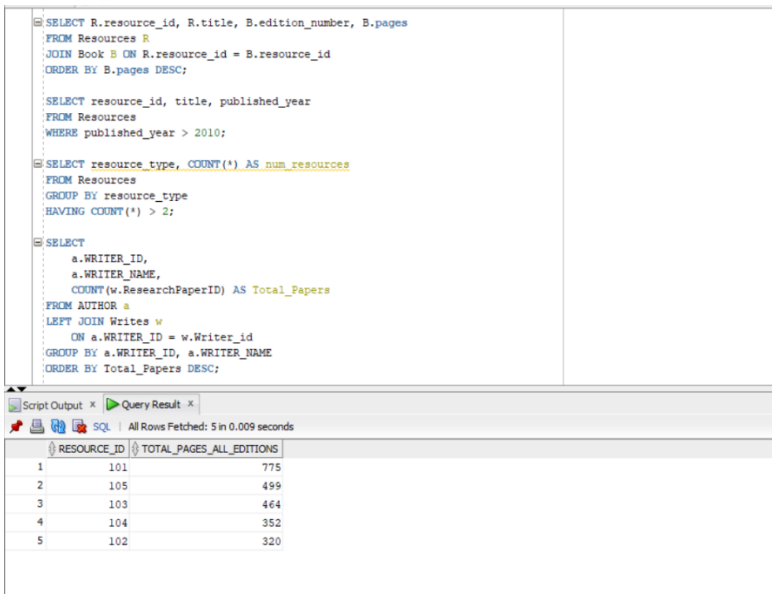
- Sorting results in ascending \descending order



```
SELECT resource_id, title, published_year
FROM Resources
WHERE published_year > 2010;

SELECT resource_type, COUNT(*) AS num_resources
FROM Resources
GROUP BY resource_type
HAVING COUNT(*) > 2;

SELECT
    a.WRITER_ID,
    a.WRITER_NAME,
    COUNT(w.ResearchPaperID) AS Total_Papers
FROM AUTHOR a
LEFT JOIN Writes w
    ON a.WRITER_ID = w.Writer_id
GROUP BY a.WRITER_ID, a.WRITER_NAME
ORDER BY Total_Papers DESC;
```

Query Result — All Rows Fetched: 5 in 0.003 seconds

	WRITER_ID	WRITER_NAME	TOTAL_PAPERS
1	1	Sara Writer	1
2	2	Lama Alharbi	1
3	5	Rana Fahad	1
4	4	Maha Omar	1
5	3	Nora Saleh	1

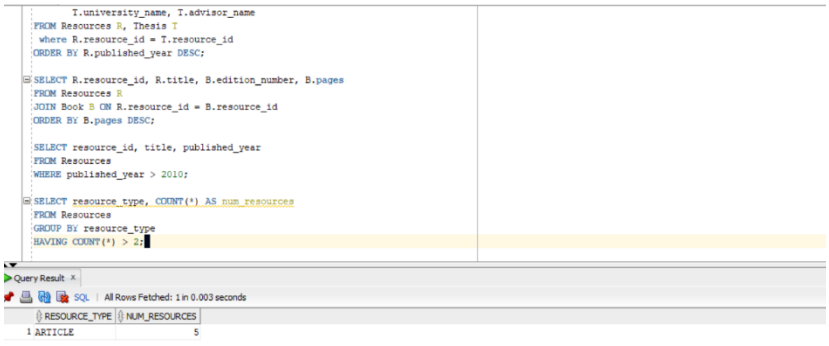


```
SELECT R.resource_id, R.title, B.edition_number, B.pages
FROM Resources R
JOIN Book B ON R.resource_id = B.resource_id
ORDER BY B.pages DESC;

SELECT resource_id, title, published_year
FROM Resources
WHERE published_year > 2010;

SELECT resource_type, COUNT(*) AS num_resources
FROM Resources
GROUP BY resource_type
HAVING COUNT(*) > 2;

SELECT
    a.WRITER_ID,
    a.WRITER_NAME,
    COUNT(w.ResearchPaperID) AS Total_Papers
FROM AUTHOR a
LEFT JOIN Writes w
    ON a.WRITER_ID = w.Writer_id
GROUP BY a.WRITER_ID, a.WRITER_NAME
ORDER BY Total_Papers DESC;
```

Script Output — Query Result — All Rows Fetched: 5 in 0.009 seconds

	RESOURCE_ID	TOTAL_PAGES_ALL_EDITIONS
1	101	775
2	105	499
3	103	464
4	104	352
5	102	320

-Aggregation and grouping(GROUP BY, HAVING)



```
    T.university_name, T.advisor_name
FROM Resources R, Thesis T
 where R.resource_id = T.resource_id
ORDER BY R.published_year DESC;

SELECT R.resource_id, R.title, B.edition_number, B.pages
FROM Resources R
JOIN Book B ON R.resource_id = B.resource_id
ORDER BY B.pages DESC;

SELECT resource_id, title, published_year
FROM Resources
WHERE published_year > 2010;

SELECT resource_type, COUNT(*) AS num_resources
FROM Resources
GROUP BY resource_type
HAVING COUNT(*) > 2;
```

Query Result — All Rows Fetched: 1 in 0.003 seconds

	RESOURCE_TYPE	NUM_RESOURCES
1	ARTICLE	5

5. Conclusion

This project provided comprehensive experience in analyzing, designing EER Model and relational models, and implementing a database system specifically designed for managing scientific and academic resources. Starting from practical requirements, the team translated operational needs—such as storing articles, books, authors, readers, and publication details—into a structured and integrated database design. The EER model allowed us to clearly visualize entities and their relationships, while the relational diagram ensured efficient and non-redundant implementation.

During the SQL implementation phase, the team deepened its understanding of data definition, constraints, and query formulation. Working with primary and foreign keys, implementing relationships, and writing DDL and DML commands helped solidify how to translate theory into a usable system.

Overall, the final system meets practical needs by offering a scalable solution capable of accommodating future growth, while effectively organizing a wide range of academic resources. In addition to fulfilling the requirements of academic projects, this work highlights how a well-designed database can enhance the experience of researchers, students, and institutions by providing structured, reliable, and easily accessible information.