M | LIBRARY

Hassan Faraj, Klevis Todi, Musa Zahoor
ISM 431 001 Database Systems II
Maggie Guo
10 December 2023

Description of Organization

The University of Michigan Library is notably a successful academic and research facility. They offer tons of thousands of resources to help students learn. Some of the resources include study halls, online applications, online news articles (New York Times, and The Washington Post), online graphics design, web development, and many more. These libraries offer many degrees for students including associates, bachelors, masters, etc. Research assistance, research consultation, course instruction, and reference support are just a few of the essential offerings provided by the University of Michigan library. As we also know The University of Michigan Hospitals are excellent hospitals due to its highly advanced research programs. The school sticks to a set of core principles that include humanity, diversity, engagement, and excellence. The University of Michigan has a known reputation for shaping the future and teaching its students to prepare them for their futures. We chose the University of Michigan for this project due to its well-known reputation for offering materials that support scholarly development, advancement, and research.

Information Requirements of Organization

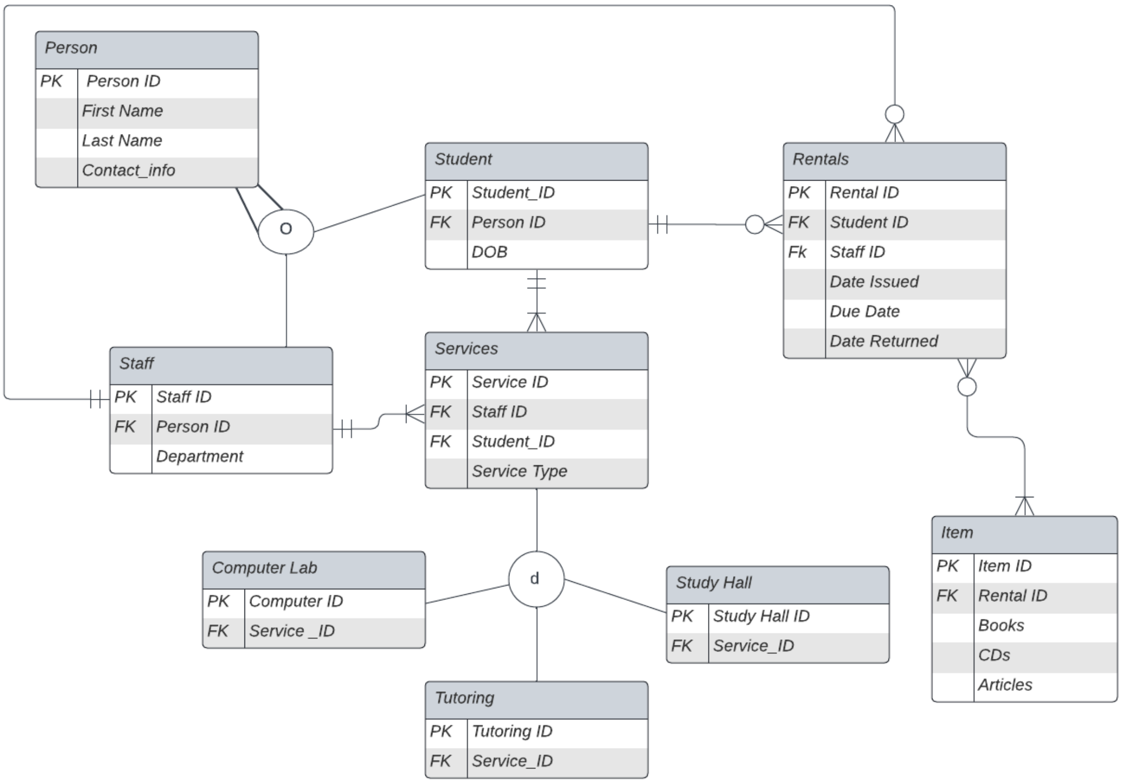
With a wide range of resources and services for research, study, development, and knowledge, the University of Michigan Library is committed to serving the various information needs of faculty and students. In keeping with the research-focused nature of the university, a primarily quantitative approach to information gathering was chosen. This database is an asset for students and teachers equally, providing a wide range of materials including books, movies, and online references in addition to services like tutoring, computer lab use, and book clubs. Every source of information is carefully organized in our database.

A wide range of services are available to users, including staff and students, from the library. Every service has a minimum of a single staff member or student associated with it. While a given book may only be borrowed by one student, students can choose to borrow zero to many books. A complete library catalog should have a minimum of one book, one movie, a single online reference.

Conceptual Model

Below we have included a Conceptual Model for the University of Michigan Library. This model outlines the structures of the tables as well as the relationships within the University of Michigan Library. We have several entities some of which include ‘Student’, ‘Person’, ‘Staff’, as well as ‘Tutoring’. Each of these entities are made up of foreign and primary keys. This ERD diagram illustrates an institution's database, with "Person" serving as the central entity and linkages to "Student" and "Staff," implying their respective functions. There is a system of renting in place when "Students" and "Staff" can engage in "Rentals" of "Items." The "Services" provided by the "Staff" are "Tutoring," "Study Hall," and "Computer Lab," which connect work positions with service locations. "Items" classify as Articles, CDs, or Books, most likely for rental use. This framework keeps track of individuals, positions, services, and item transactions for an extensive database.

The Person table overlaps with the Staff and Student tables because a person can be both a staff and student. Staff table to Rentals table: one and only one staff can help facilitate zero or many rentals. Staff table to Services table: one and only one staff can provide one or many services. Student table to Rentals table: one and only one student can rent zero or many items. Rentals table to Items table: zero or many rentals for one or many items. Student table to Services table: one and only one student can utilize one or many of the services provided. Services table disjoints with Computer Lab, Tutoring, and Study Hall table because a student can only use one service at a time.



Set of Normalized Relations

PK is **Bolded**

FK is *Italicized*

Person (**person_id**, first_name, last_name, Contact_info)
Student(**student_id**, *person_id*, DOB)
Staff(**staff_id**, *person_id*, Department)
Rentals(**rental_id**, *student_id*, *staff_id*, date_issued, due_date, date_returned)

Services(**service_id**, *staff_id*, *student_id*, service_type)
Computer Lab (**computer_id**, *service_id*)
Tutoring (**Tutoring_id**, *service_id*)
Study Hall (**study_hall_id**, *service_id*)
Item (**Item_id**, *Rental_id*, books, CDs, Articles)

Physical Design

Student

Column	**Datatype**	**Constraint?**	**Default Values?**
Student_ID	char(10)	Primary key	no
Person_ID	char(10)	Foreign key refers back to person_id in person	no
DOB	Date		no

Person

Column	**Datatype**	**Constraint?**	**Default Values?**
person_ID	char(10)	Primary key	no
name	varchar2(20)	Not null	no
contact_info	varchar2(60)	Not null	No

Staff

Column	Datatype	Constraint?	Default Values?
staff_id	char(10)	Primary key	no
person_id	char(10)	Foreign key refers back to person_id in person	no
Department	varchar2(20)		no

Rentals

Column	Datatype	Constraint?	Default Values?
Rental_id	char(10)	Primary key	no
student_id	char(10)	Foreign key refers back to student_id in student	no
staff_id	char(10)	Foreign key refers back to staff_id in staff	no
Date_Issued	date		no
Due_Date	date		no
Date_Returned	date		no

Services

Column	Datatype	Constraint?	Default Values?
Service_id	char(10)	Primary key	no
Staff_id	char(10)	Foreign key refers back to staff_id in staff	no
student_id	char(10)	Foreign key refers back to student_id in student	no
Service type	varchar2(200)		no

Computer Lab

Column	Datatype	Constraint?	Default Values?

computer_id	char(10)	Primary key	no
service_id	char(10)	Foreign key refers back to service_id in services	no

Tutoring

Column	Datatype	Constraint?	Default Values?
Tutoring_id	char(10)	Primary key	no
service_id	char(10)	Foreign key refers back to service_id in services	no

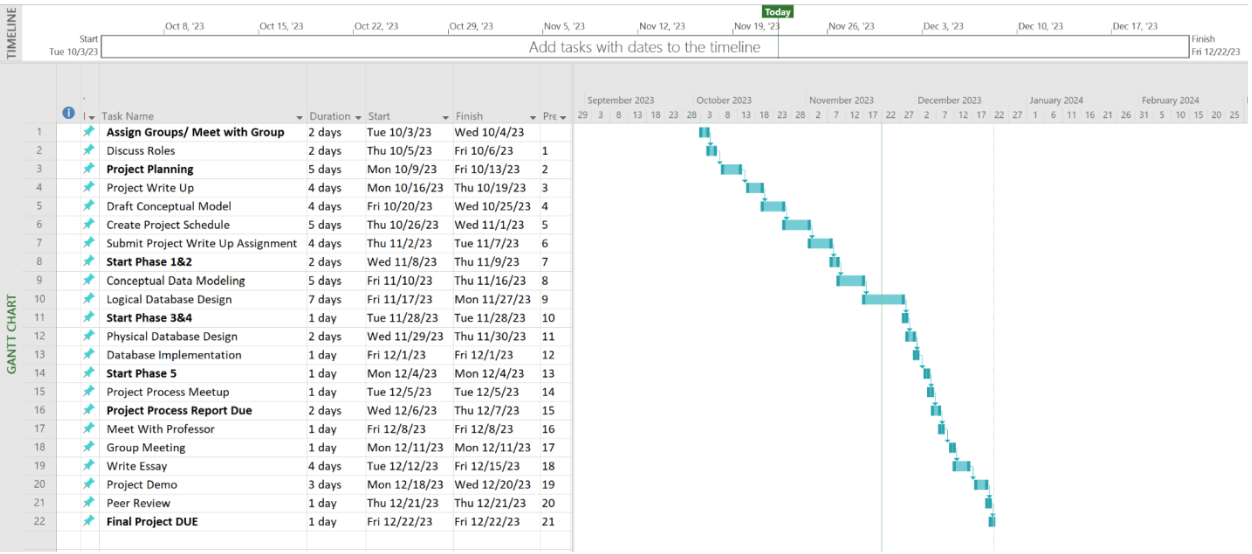
Study Hall

Column	Datatype	Constraint?	Default Values?
study_hall_id	char(10)	Primary key	no
service_id	char(10)	Foreign key refers back to service_id in services	no

Items

Column	Datatype	Constraint?	Default Values?
Item_id	char(10)	Primary key	no
rental_id	char(10)	Foreign key refers back to rental_id in rentals	no
Books	varchar2(200)		no
cds	varchar2(200)		no
Articles	varchar2(200)		no

Microsoft Project - Project Schedule



	Task Name	Duration	Start	Finish	Pre
1	**Assign Groups/ Meet with Group**	2 days	Tue 10/3/23	Wed 10/4/23	
2	Discuss Roles	2 days	Thu 10/5/23	Fri 10/6/23	1
3	**Project Planning**	5 days	Mon 10/9/23	Fri 10/13/23	2
4	Project Write Up	4 days	Mon 10/16/23	Thu 10/19/23	3
5	Draft Conceptual Model	4 days	Fri 10/20/23	Wed 10/25/23	4
6	Create Project Schedule	5 days	Thu 10/26/23	Wed 11/1/23	5
7	Submit Project Write Up Assignment	4 days	Thu 11/2/23	Tue 11/7/23	6
8	**Start Phase 1&2**	2 days	Wed 11/8/23	Thu 11/9/23	7
9	Conceptual Data Modeling	5 days	Fri 11/10/23	Thu 11/16/23	8
10	Logical Database Design	7 days	Fri 11/17/23	Mon 11/27/23	9
11	**Start Phase 3&4**	1 day	Tue 11/28/23	Tue 11/28/23	10
12	Physical Database Design	2 days	Wed 11/29/23	Thu 11/30/23	11
13	Database Implementation	1 day	Fri 12/1/23	Fri 12/1/23	12
14	**Start Phase 5**	1 day	Mon 12/4/23	Mon 12/4/23	13
15	Project Process Meetup	1 day	Tue 12/5/23	Tue 12/5/23	14
16	**Project Process Report Due**	2 days	Wed 12/6/23	Thu 12/7/23	15
17	Meet With Professor	1 day	Fri 12/8/23	Fri 12/8/23	16
18	Group Meeting	1 day	Mon 12/11/23	Mon 12/11/23	17
19	Write Essay	4 days	Tue 12/12/23	Fri 12/15/23	18
20	Project Demo	3 days	Mon 12/18/23	Wed 12/20/23	19
21	Peer Review	1 day	Thu 12/21/23	Thu 12/21/23	20
22	**Final Project DUE**	1 day	Fri 12/22/23	Fri 12/22/23	21

Sign Off Sheet

Klevis Todi: Klevis worked on the conceptual model, ERD, Physical design. He also typed the background paragraph and microsoft project.

Hassan Faraj: Hassan worked on the physical design as well as making the queries for it. He helped describe the information in the information paragraph. Worked on the SQL code.

Musa Zahoor: Musa worked on the project schedule, queries, and conceptual design. He also added information requirements.

Journal of Meetings

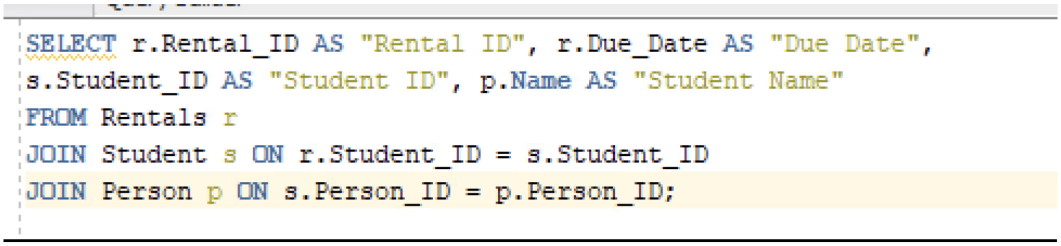
- October 3- Group Chat created
- October 5 - First meeting/ Discuss roles
- October 9 - Project Planning

- October 10 - Zoom Meeting. Discussed project plan write-up
- October 15 - Zoom Meeting. Created project plan
- October 17 - Zoom meeting. Drafted conceptual model.
- November 6- Met after class. Fix Conceptual Model. Discuss Logical Model.
- November 12- Finalize Logical Model.
- November 15- Zoom meeting, Write SQL script and make sure we all understand everything.
- December 1- Finalize SQL Script, Create all queries.
- December 8- Create Presentation, finalize project, and edit everything.

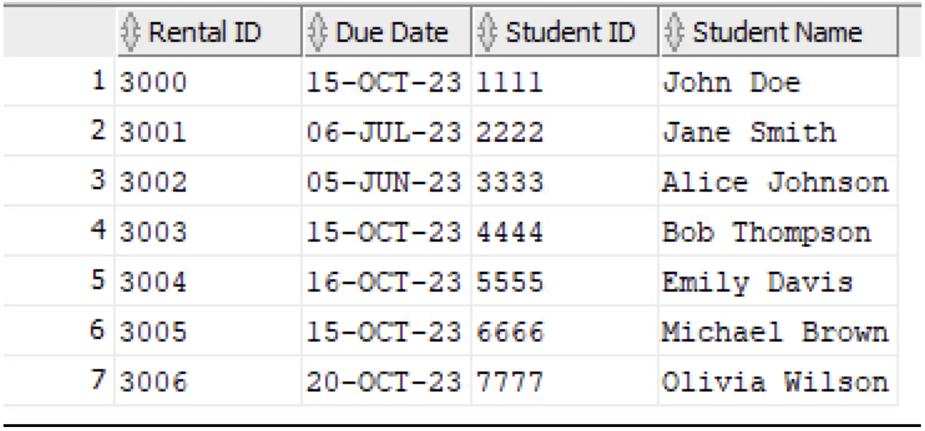
Signed by: Hassan Faraj, Musa Zahoor, Klevis Todi

Documentation of SQL Queries

(Column aliases):



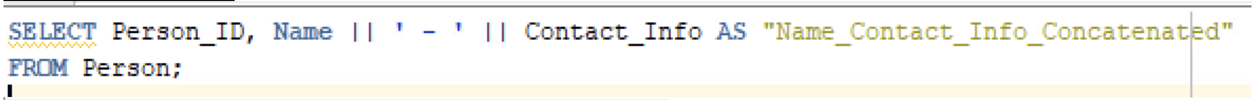
```
SELECT r.Rental_ID AS "Rental ID", r.Due_Date AS "Due Date",
s.Student_ID AS "Student ID", p.Name AS "Student Name"
FROM Rentals r
JOIN Student s ON r.Student_ID = s.Student_ID
JOIN Person p ON s.Person_ID = p.Person_ID;
```



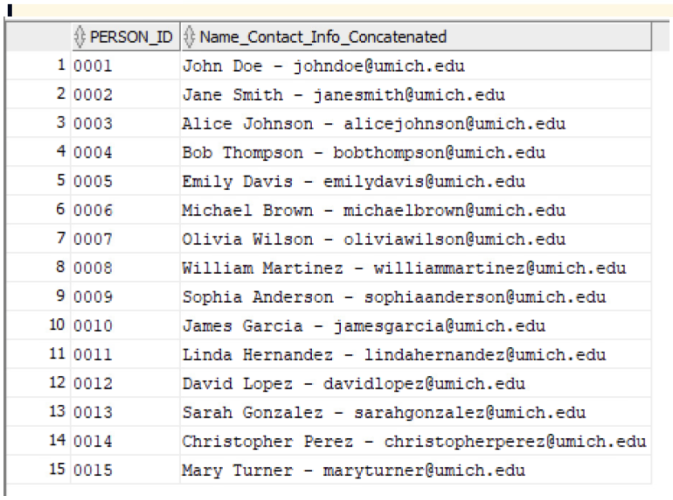
	Rental ID	Due Date	Student ID	Student Name
1	3000	15-OCT-23	1111	John Doe
2	3001	06-JUL-23	2222	Jane Smith
3	3002	05-JUN-23	3333	Alice Johnson
4	3003	15-OCT-23	4444	Bob Thompson
5	3004	16-OCT-23	5555	Emily Davis
6	3005	15-OCT-23	6666	Michael Brown
7	3006	20-OCT-23	7777	Olivia Wilson

The purpose of the query above is to retrieve information about rentals, including the rental ID, due date, student ID, and the name of the student.

(Concatenation):



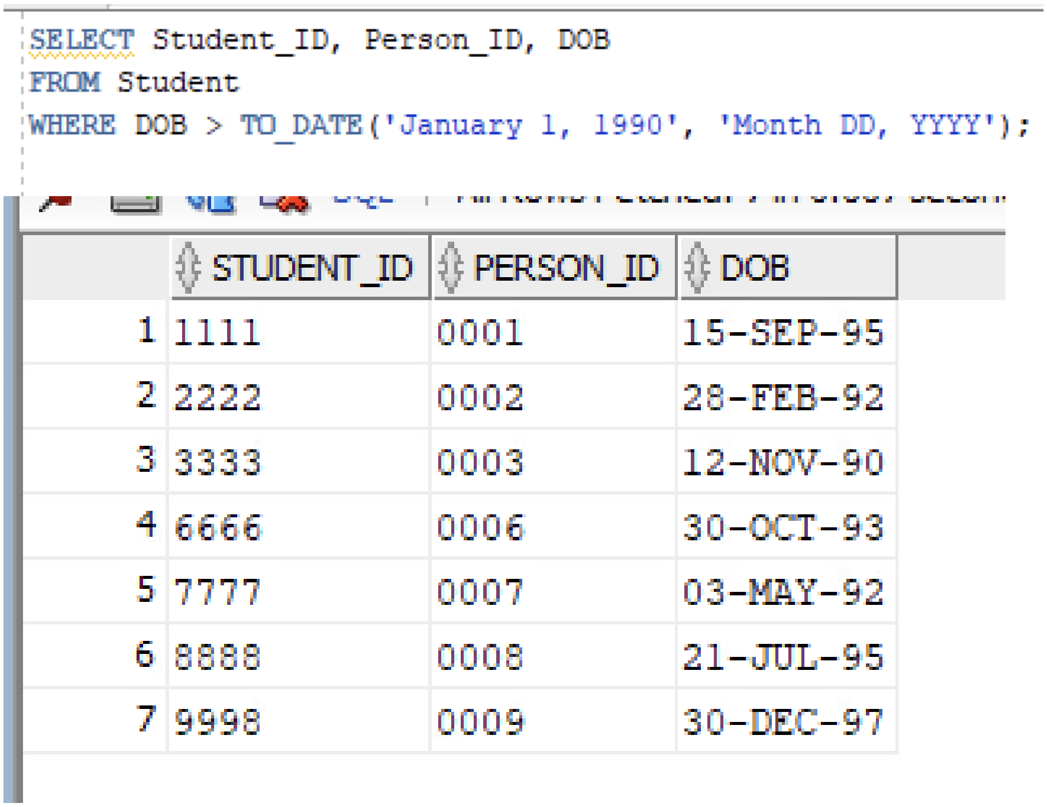
```
SELECT Person_ID, Name || ' - ' || Contact_Info AS "Name_Contact_Info_Concatenated"
FROM Person;
```



	PERSON_ID	Name_Contact_Info_Concatenated
1	0001	John Doe - johndoe@umich.edu
2	0002	Jane Smith - janesmith@umich.edu
3	0003	Alice Johnson - alicejohnson@umich.edu
4	0004	Bob Thompson - bobthompson@umich.edu
5	0005	Emily Davis - emilydavis@umich.edu
6	0006	Michael Brown - michaelbrown@umich.edu
7	0007	Olivia Wilson - oliviawilson@umich.edu
8	0008	William Martinez - williammartinez@umich.edu
9	0009	Sophia Anderson - sophiaanderson@umich.edu
10	0010	James Garcia - jamesgarcia@umich.edu
11	0011	Linda Hernandez - lindahernandez@umich.edu
12	0012	David Lopez - davidlopez@umich.edu
13	0013	Sarah Gonzalez - sarahgonzalez@umich.edu
14	0014	Christopher Perez - christopherperez@umich.edu
15	0015	Mary Turner - maryturner@umich.edu

The purpose of this query above is to is to concatenate the Name and Contact_Info

(Comparison Operators):

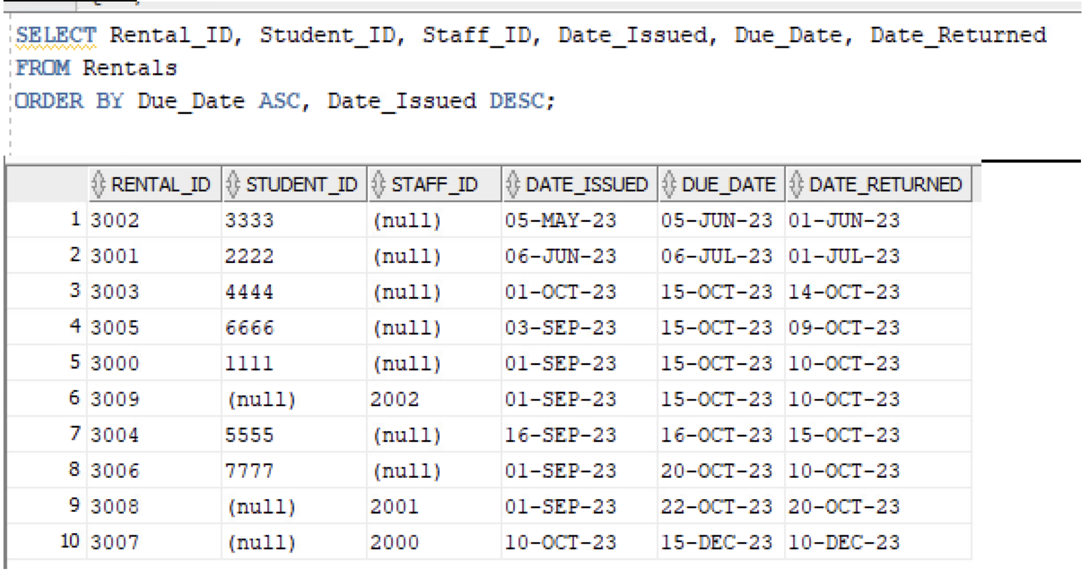


```
SELECT Student_ID, Person_ID, DOB
FROM Student
WHERE DOB > TO_DATE('January 1, 1990', 'Month DD, YYYY');
```

	STUDENT_ID	PERSON_ID	DOB
1	1111	0001	15-SEP-95
2	2222	0002	28-FEB-92
3	3333	0003	12-NOV-90
4	6666	0006	30-OCT-93
5	7777	0007	03-MAY-92
6	8888	0008	21-JUL-95
7	9998	0009	30-DEC-97

The purpose of the query above is to to retrieve information about students who were born after January 1, 1990

(Sorting):



```
SELECT Rental_ID, Student_ID, Staff_ID, Date_Issued, Due_Date, Date_Returned
FROM Rentals
ORDER BY Due_Date ASC, Date_Issued DESC;
```

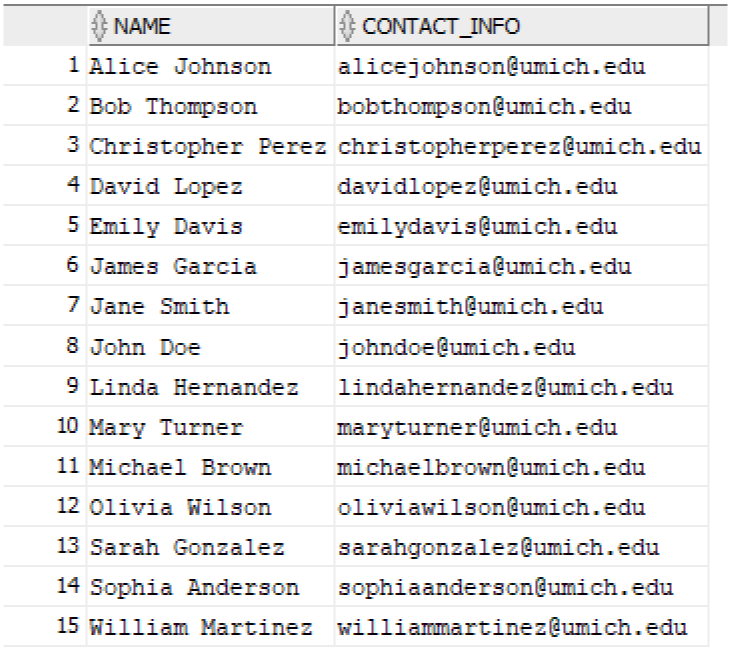
	RENTAL_ID	STUDENT_ID	STAFF_ID	DATE_ISSUED	DUE_DATE	DATE_RETURNED
1	3002	3333	(null)	05-MAY-23	05-JUN-23	01-JUN-23
2	3001	2222	(null)	06-JUN-23	06-JUL-23	01-JUL-23
3	3003	4444	(null)	01-OCT-23	15-OCT-23	14-OCT-23
4	3005	6666	(null)	03-SEP-23	15-OCT-23	09-OCT-23
5	3000	1111	(null)	01-SEP-23	15-OCT-23	10-OCT-23
6	3009	(null)	2002	01-SEP-23	15-OCT-23	10-OCT-23
7	3004	5555	(null)	16-SEP-23	16-OCT-23	15-OCT-23
8	3006	7777	(null)	01-SEP-23	20-OCT-23	10-OCT-23
9	3008	(null)	2001	01-SEP-23	22-OCT-23	20-OCT-23
10	3007	(null)	2000	10-OCT-23	15-DEC-23	10-DEC-23

The purpose of the query above is to sort the result set first by the "Due_Date" column in ascending order and then by the "Date_Issued" column in descending order.

SQL Statement: SELECT Name, Contact_Info
FROM Person

ORDER BY Name;

Purpose: The purpose of this first query is to show the contact information of all students and staff in alphabetical order. This helps in organizing information of people in our database and makes it easier to look someone up.



	NAME	CONTACT_INFO
1	Alice Johnson	alicejohnson@umich.edu
2	Bob Thompson	bobthompson@umich.edu
3	Christopher Perez	christopherperez@umich.edu
4	David Lopez	davidlopez@umich.edu
5	Emily Davis	emilydavis@umich.edu
6	James Garcia	jamesgarcia@umich.edu
7	Jane Smith	janesmith@umich.edu
8	John Doe	johndoe@umich.edu
9	Linda Hernandez	lindahernandez@umich.edu
10	Mary Turner	maryturner@umich.edu
11	Michael Brown	michaelbrown@umich.edu
12	Olivia Wilson	oliviawilson@umich.edu
13	Sarah Gonzalez	sarahgonzalez@umich.edu
14	Sophia Anderson	sophiaanderson@umich.edu
15	William Martinez	williammartinez@umich.edu

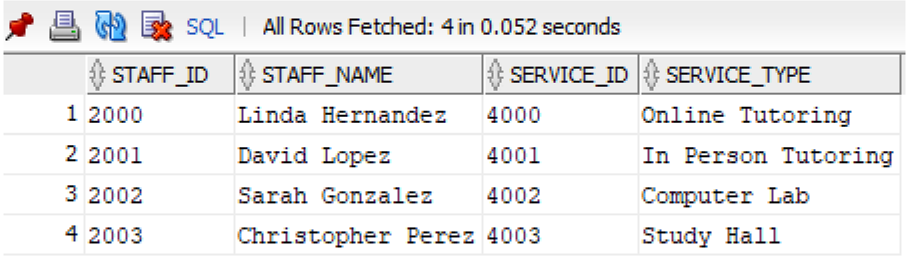
(Joining Tables)

SQL Statement: SELECT Staff.Staff_ID, Person.Name AS Staff_Name, Services.Service_ID, Services.Service_Type FROM Staff

JOIN Services ON Staff.Staff_ID = Services.Staff_ID

JOIN Person ON Staff.Person_ID = Person.Person_ID;

Purpose: This query answers the question: "Which staff members (along with their names) are associated with what services?" The query JOINS multiple tables to show what services our staff members provide.



All Rows Fetched: 4 in 0.052 seconds

	STAFF_ID	STAFF_NAME	SERVICE_ID	SERVICE_TYPE
1	2000	Linda Hernandez	4000	Online Tutoring
2	2001	David Lopez	4001	In Person Tutoring
3	2002	Sarah Gonzalez	4002	Computer Lab
4	2003	Christopher Perez	4003	Study Hall

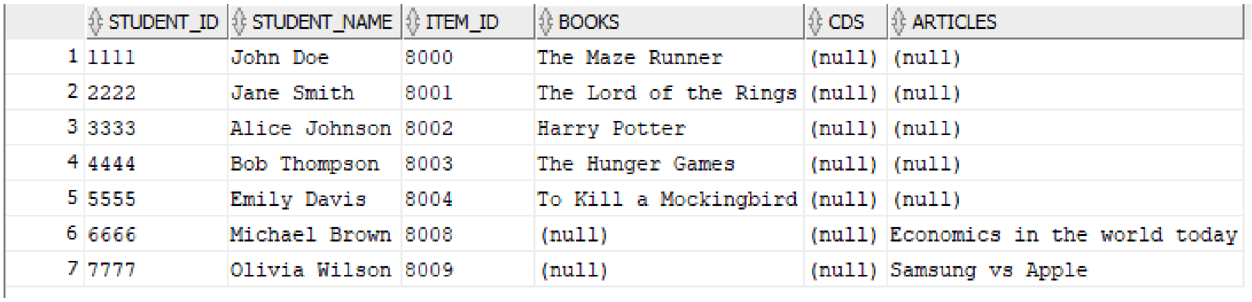
SQL Statement: SELECT Student.Student_ID, Person.Name AS Student_Name, Items.Item_ID,Items.Books,Items.CDs,Items.Articles FROM Student

JOIN Rentals ON Student.Student_ID = Rentals.Student_ID

JOIN Items ON Rentals.Rental_ID = Items.Rental_ID

JOIN Person ON Student.Person_ID = Person.Person_ID;

Purpose: This query answers the question: “What specific items (Books, CDs, Articles) are rented by each student, along with their associated details?” This query retrieves details about the items rented by students, including the identification of students. The null values show that no item was rented for that student.



	STUDENT_ID	STUDENT_NAME	ITEM_ID	BOOKS	CDS	ARTICLES
1	1111	John Doe	8000	The Maze Runner	(null)	(null)
2	2222	Jane Smith	8001	The Lord of the Rings	(null)	(null)
3	3333	Alice Johnson	8002	Harry Potter	(null)	(null)
4	4444	Bob Thompson	8003	The Hunger Games	(null)	(null)
5	5555	Emily Davis	8004	To Kill a Mockingbird	(null)	(null)
6	6666	Michael Brown	8008	(null)	(null)	Economics in the world today
7	7777	Olivia Wilson	8009	(null)	(null)	Samsung vs Apple

(Case Conversion Functions)

SQL Statement: SELECT Date_Issued,Due_Date,Date_Returned,**CASE**

WHEN Date_Returned IS NULL THEN 'Not Returned'

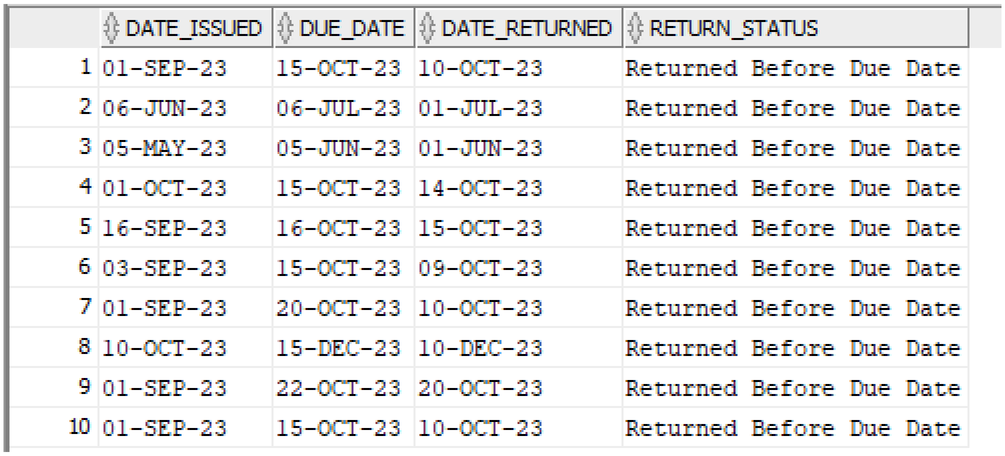
WHEN Date_Returned <= Due_Date THEN 'Returned Before Due Date'

ELSE 'Returned After Due Date'

END AS Return_Status

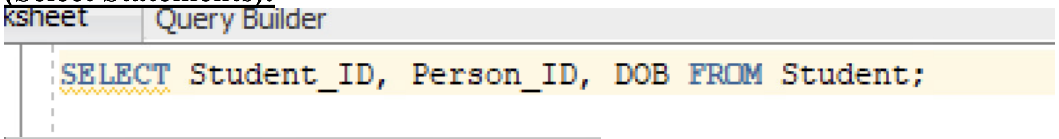
FROM Rentals;

Purpose: This query answers the question: “What is the status of the return for each rental item based on its due date and return date?” This query uses a ‘CASE’ statement to evaluate whether the items were returned before the due date, after the due date, or not returned at all.

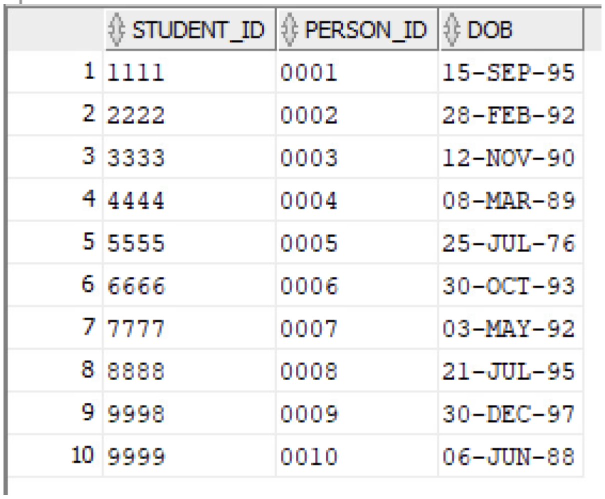


	DATE_ISSUED	DUE_DATE	DATE_RETURNED	RETURN_STATUS
1	01-SEP-23	15-OCT-23	10-OCT-23	Returned Before Due Date
2	06-JUN-23	06-JUL-23	01-JUL-23	Returned Before Due Date
3	05-MAY-23	05-JUN-23	01-JUN-23	Returned Before Due Date
4	01-OCT-23	15-OCT-23	14-OCT-23	Returned Before Due Date
5	16-SEP-23	16-OCT-23	15-OCT-23	Returned Before Due Date
6	03-SEP-23	15-OCT-23	09-OCT-23	Returned Before Due Date
7	01-SEP-23	20-OCT-23	10-OCT-23	Returned Before Due Date
8	10-OCT-23	15-DEC-23	10-DEC-23	Returned Before Due Date
9	01-SEP-23	22-OCT-23	20-OCT-23	Returned Before Due Date
10	01-SEP-23	15-OCT-23	10-OCT-23	Returned Before Due Date

(Select Statements):

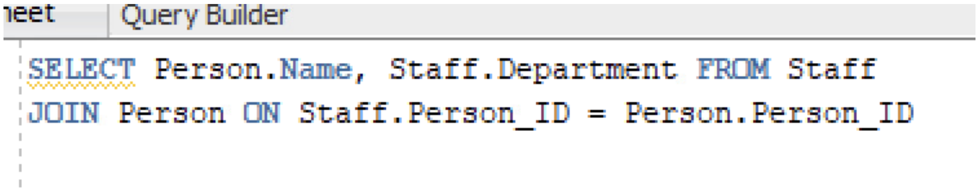


```
SELECT Student_ID, Person_ID, DOB FROM Student;
```

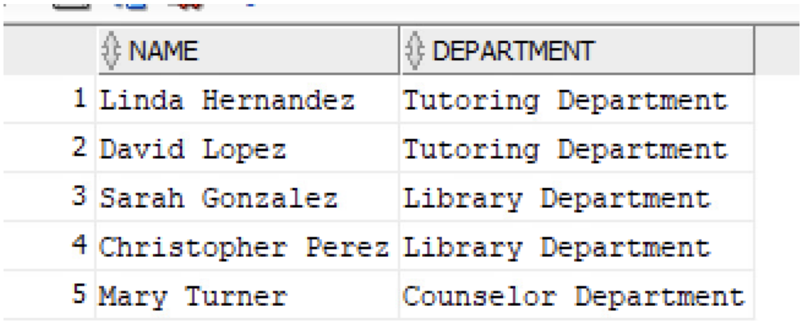


	STUDENT_ID	PERSON_ID	DOB
1	1111	0001	15-SEP-95
2	2222	0002	28-FEB-92
3	3333	0003	12-NOV-90
4	4444	0004	08-MAR-89
5	5555	0005	25-JUL-76
6	6666	0006	30-OCT-93
7	7777	0007	03-MAY-92
8	8888	0008	21-JUL-95
9	9998	0009	30-DEC-97
10	9999	0010	06-JUN-88

The query above selects Student_ID, Person_ID and the DOB from the students table.

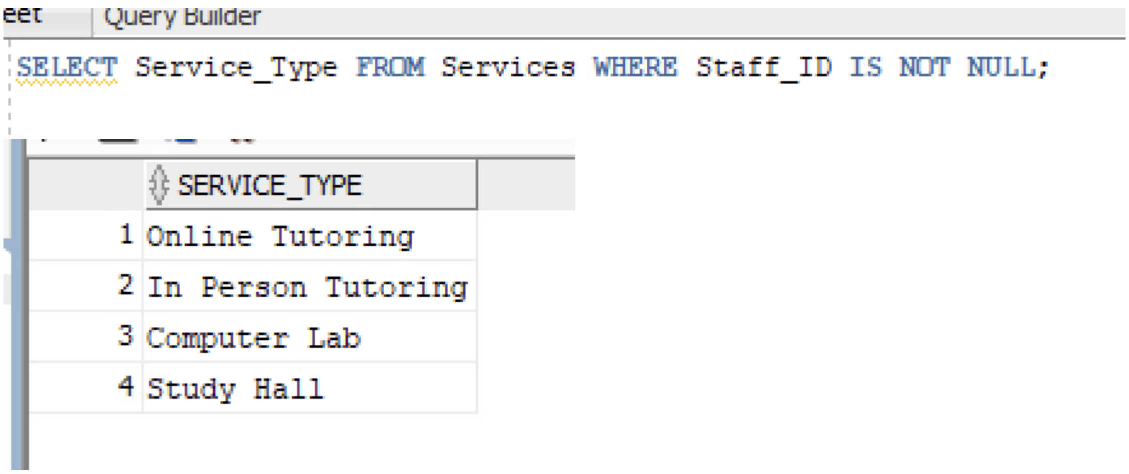


```
SELECT Person.Name, Staff.Department FROM Staff
JOIN Person ON Staff.Person_ID = Person.Person_ID
```



	NAME	DEPARTMENT
1	Linda Hernandez	Tutoring Department
2	David Lopez	Tutoring Department
3	Sarah Gonzalez	Library Department
4	Christopher Perez	Library Department
5	Mary Turner	Counselor Department

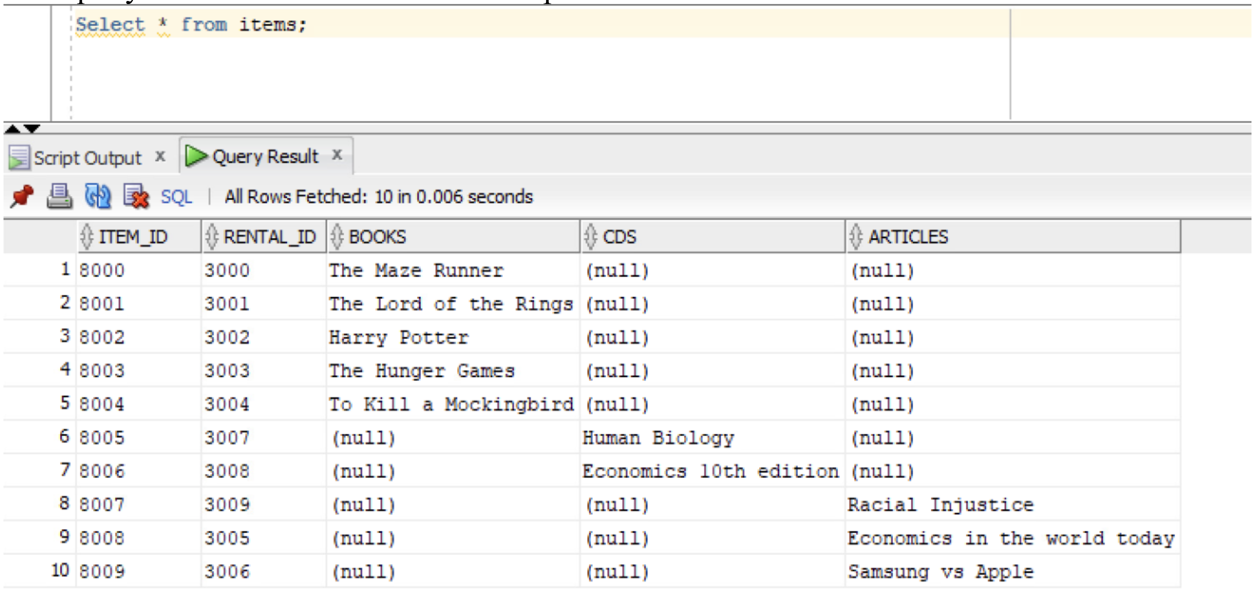
The query above selects the name of the staff members as well as the department they are part of.



```
SELECT Service_Type FROM Services WHERE Staff_ID IS NOT NULL;
```

	SERVICE_TYPE
1	Online Tutoring
2	In Person Tutoring
3	Computer Lab
4	Study Hall

This query selects all the services that are provided.

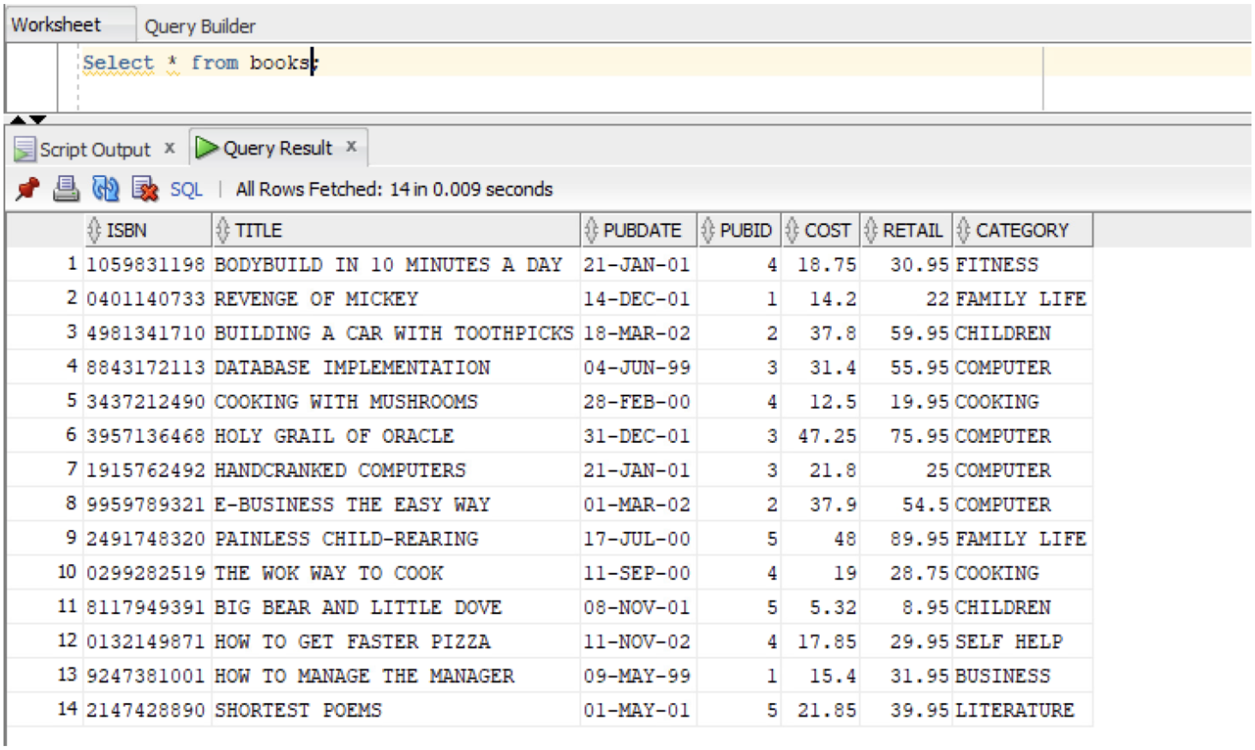


```
Select * from items;
```

All Rows Fetched: 10 in 0.006 seconds

	ITEM_ID	RENTAL_ID	BOOKS	CDS	ARTICLES
1	8000	3000	The Maze Runner	(null)	(null)
2	8001	3001	The Lord of the Rings	(null)	(null)
3	8002	3002	Harry Potter	(null)	(null)
4	8003	3003	The Hunger Games	(null)	(null)
5	8004	3004	To Kill a Mockingbird	(null)	(null)
6	8005	3007	(null)	Human Biology	(null)
7	8006	3008	(null)	Economics 10th edition	(null)
8	8007	3009	(null)	(null)	Racial Injustice
9	8008	3005	(null)	(null)	Economics in the world today
10	8009	3006	(null)	(null)	Samsung vs Apple

This query shows all the items in the library that can be rented.

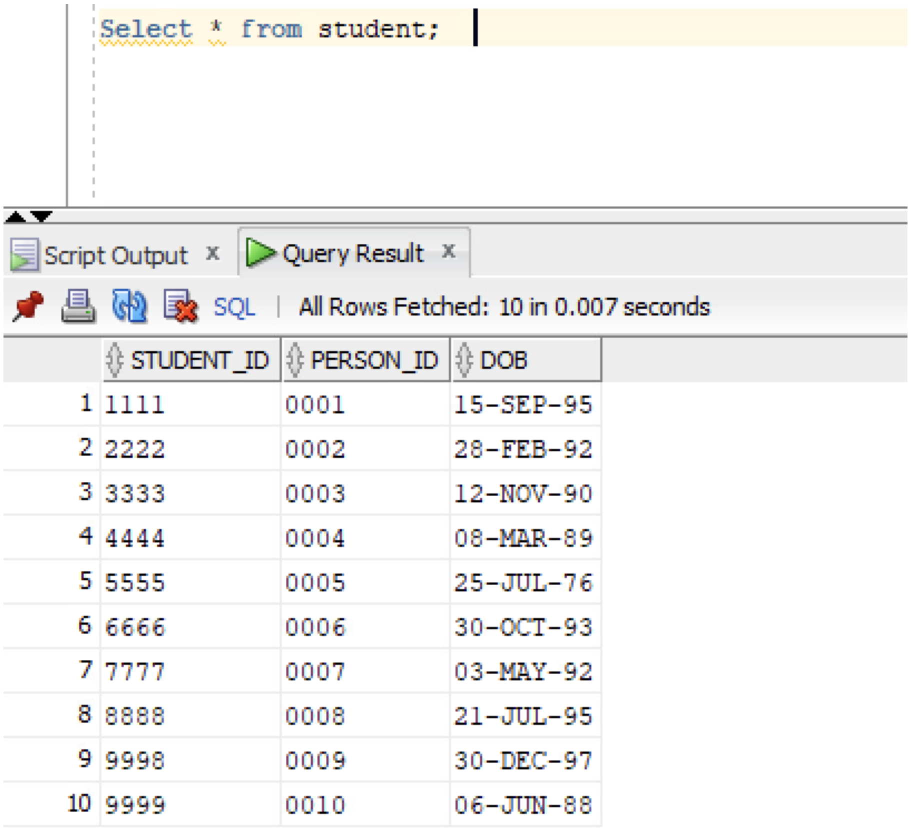


```
Select * from books;
```

All Rows Fetched: 14 in 0.009 seconds

	ISBN	TITLE	PUBDATE	PUBID	COST	RETAIL	CATEGORY
1	1059831198	BODYBUILD IN 10 MINUTES A DAY	21-JAN-01	4	18.75	30.95	FITNESS
2	0401140733	REVENGE OF MICKEY	14-DEC-01	1	14.2	22	FAMILY LIFE
3	4981341710	BUILDING A CAR WITH TOOTHPICKS	18-MAR-02	2	37.8	59.95	CHILDREN
4	8843172113	DATABASE IMPLEMENTATION	04-JUN-99	3	31.4	55.95	COMPUTER
5	3437212490	COOKING WITH MUSHROOMS	28-FEB-00	4	12.5	19.95	COOKING
6	3957136468	HOLY GRAIL OF ORACLE	31-DEC-01	3	47.25	75.95	COMPUTER
7	1915762492	HANDCRANKED COMPUTERS	21-JAN-01	3	21.8	25	COMPUTER
8	9959789321	E-BUSINESS THE EASY WAY	01-MAR-02	2	37.9	54.5	COMPUTER
9	2491748320	PAINLESS CHILD-REARING	17-JUL-00	5	48	89.95	FAMILY LIFE
10	0299282519	THE WOK WAY TO COOK	11-SEP-00	4	19	28.75	COOKING
11	8117949391	BIG BEAR AND LITTLE DOVE	08-NOV-01	5	5.32	8.95	CHILDREN
12	0132149871	HOW TO GET FASTER PIZZA	11-NOV-02	4	17.85	29.95	SELF HELP
13	9247381001	HOW TO MANAGE THE MANAGER	09-MAY-99	1	15.4	31.95	BUSINESS
14	2147428890	SHORTEST POEMS	01-MAY-01	5	21.85	39.95	LITERATURE

This Query shows all the books in the library.

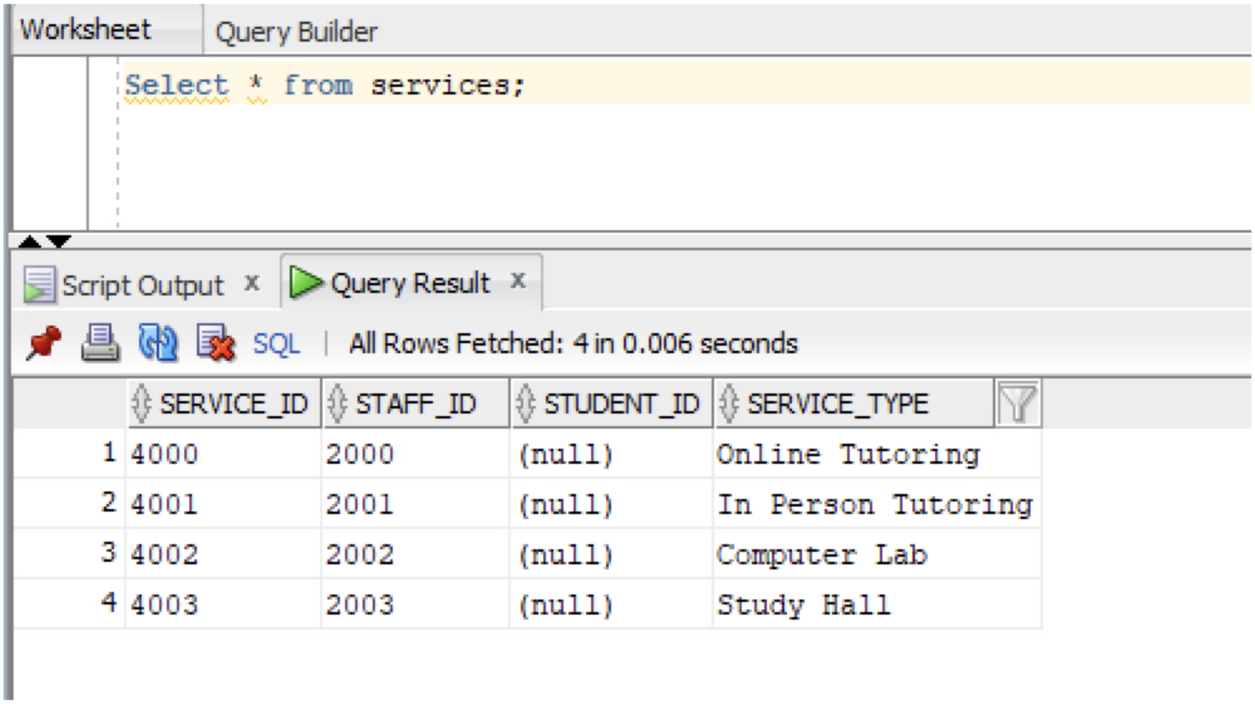


```
Select * from student;
```

All Rows Fetched: 10 in 0.007 seconds

```
    STUDENT_ID  PERSON_ID  DOB
 1  1111        0001       15-SEP-95
 2  2222        0002       28-FEB-92
 3  3333        0003       12-NOV-90
 4  4444        0004       08-MAR-89
 5  5555        0005       25-JUL-76
 6  6666        0006       30-OCT-93
 7  7777        0007       03-MAY-92
 8  8888        0008       21-JUL-95
 9  9998        0009       30-DEC-97
10  9999        0010       06-JUN-88
```

This query shows all the information on the student table.

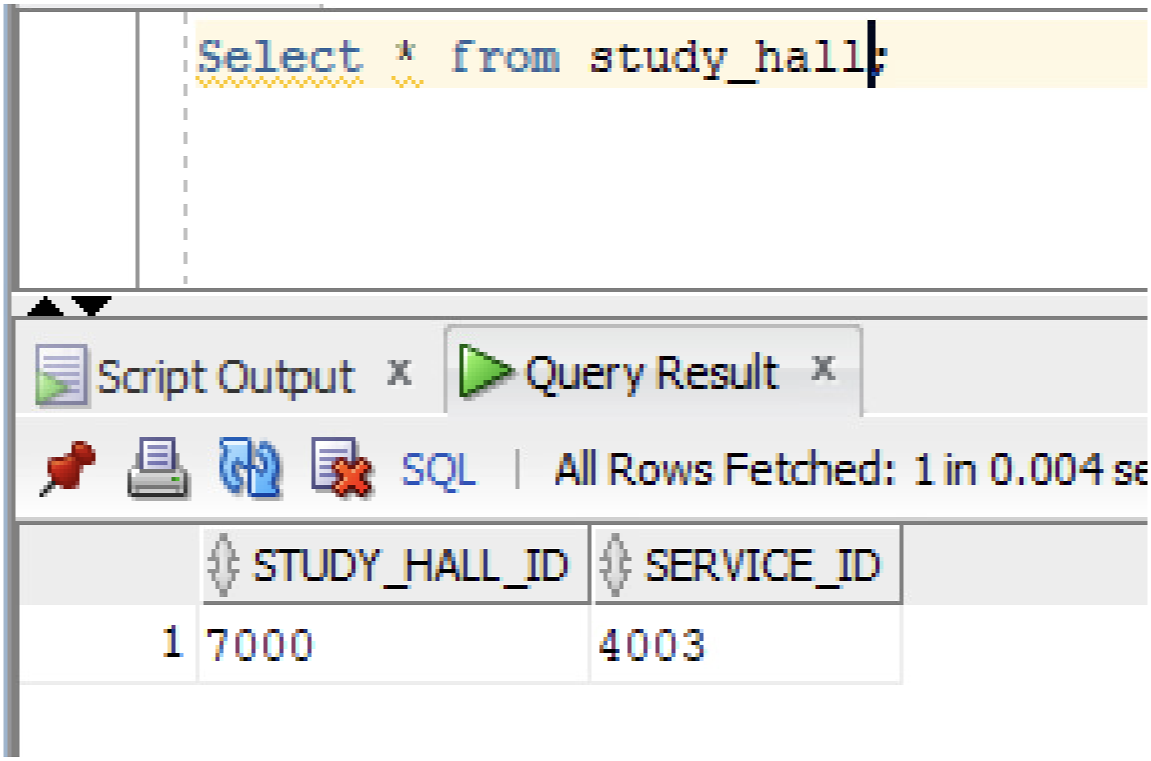


```
Select * from services;
```

All Rows Fetched: 4 in 0.006 seconds

	SERVICE_ID	STAFF_ID	STUDENT_ID	SERVICE_TYPE
1	4000	2000	(null)	Online Tutoring
2	4001	2001	(null)	In Person Tutoring
3	4002	2002	(null)	Computer Lab
4	4003	2003	(null)	Study Hall

This query shows all of the services provided by University of Michigan Library.



```
Select * from study_hall;
```

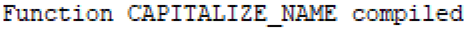
All Rows Fetched: 1 in 0.004 se

	STUDY_HALL_ID	SERVICE_ID
1	7000	4003

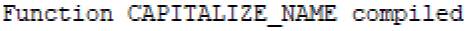
This query shows all the information in the study hall table.

Character Manipulation functions:

SQL Satement: CREATE OR REPLACE FUNCTION capitalize_name(name IN VARCHAR2)
RETURN VARCHAR2 IS
capitalized_name VARCHAR2(50); BEGIN
capitalized_name := INITCAP(name);
RETURN capitalized_name;
END;
Purpose: Function to Capitalize Names

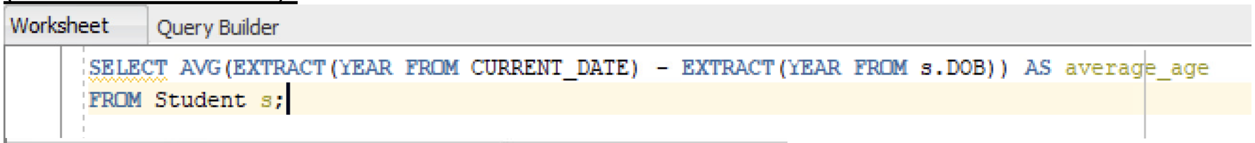


```
Function CAPITALIZE_NAME compiled
```

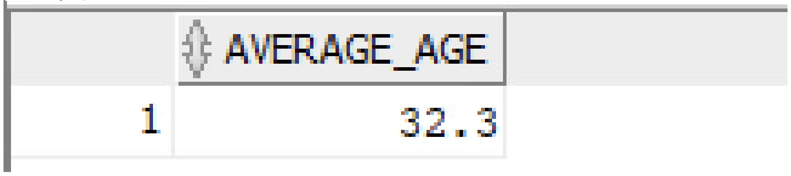


```
Function CAPITALIZE_NAME compiled
```

(Numeric Functions):



```
SELECT AVG(EXTRACT(YEAR FROM CURRENT_DATE) - EXTRACT(YEAR FROM s.DOB)) AS average_age
FROM Student s;
```

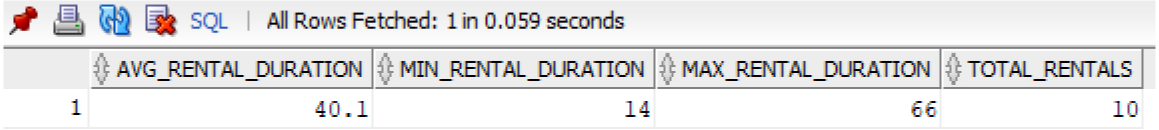


	AVERAGE_AGE
1	32.3

The purpose of the query above is to find the average age of library renters.

SQL Statement: SELECT
AVG(Due_Date - Date_Issued) AS Avg_Rental_Duration,
MIN(Due_Date - Date_Issued) AS Min_Rental_Duration,
MAX(Due_Date - Date_Issued) AS Max_Rental_Duration,
COUNT(*) AS Total_Rentals
FROM Rentals;

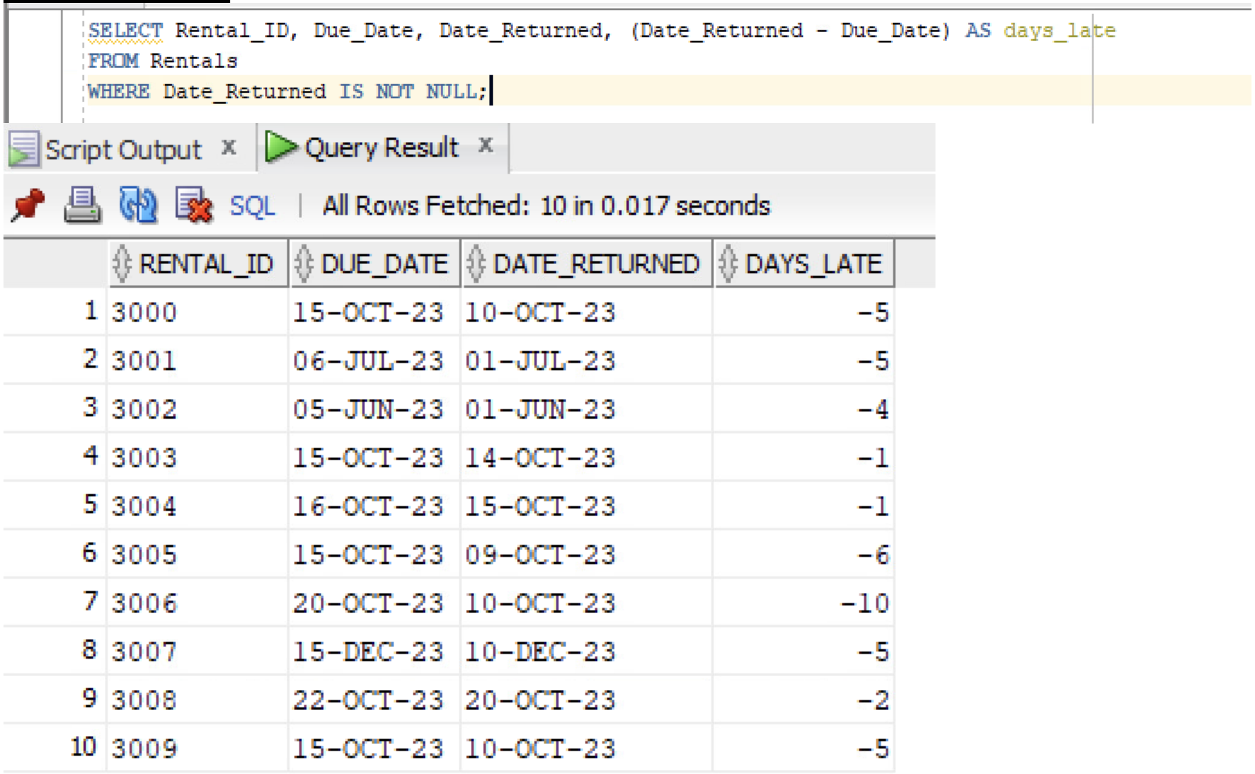
Purpose: This query answers the question: “ What is the average, minimum, and maximum duration that students or staff rent items? The query shows how long students or staff rent items for and the total amount of items rented at this point in time.



All Rows Fetched: 1 in 0.059 seconds

	AVG_RENTAL_DURATION	MIN_RENTAL_DURATION	MAX_RENTAL_DURATION	TOTAL_RENTALS
1	40.1	14	66	10

(Date Function):



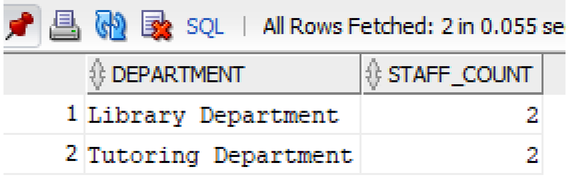
```
SELECT Rental_ID, Due_Date, Date_Returned, (Date_Returned - Due_Date) AS days_late
FROM Rentals
WHERE Date_Returned IS NOT NULL;
```

All Rows Fetched: 10 in 0.017 seconds

	RENTAL_ID	DUE_DATE	DATE_RETURNED	DAYS_LATE
1	3000	15-OCT-23	10-OCT-23	-5
2	3001	06-JUL-23	01-JUL-23	-5
3	3002	05-JUN-23	01-JUN-23	-4
4	3003	15-OCT-23	14-OCT-23	-1
5	3004	16-OCT-23	15-OCT-23	-1
6	3005	15-OCT-23	09-OCT-23	-6
7	3006	20-OCT-23	10-OCT-23	-10
8	3007	15-DEC-23	10-DEC-23	-5
9	3008	22-OCT-23	20-OCT-23	-2
10	3009	15-OCT-23	10-OCT-23	-5

The purpose of the query above is to find out how many days the person kept the material rented for.

(Group Functions)
COUNT SQL Statement: SELECT Department, COUNT(*) AS Staff_Count
FROM Staff
GROUP BY Department;
Purpose: This query counts the number of staff members in each department by grouping the data based on the Department column from the Staff table.

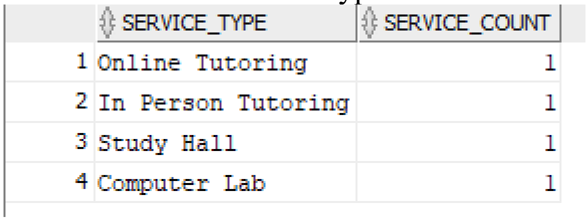


All Rows Fetched: 2 in 0.055 se

	DEPARTMENT	STAFF_COUNT
1	Library Department	2
2	Tutoring Department	2

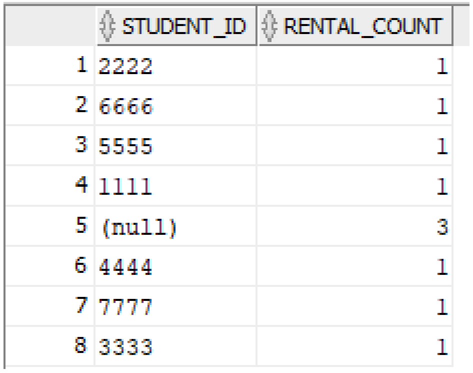
GROUP BY SQL Statement: SELECT Service_Type, COUNT(*) AS Service_Count
FROM Services

GROUP BY Service_Type;
Purpose: The query groups services based on their types from the Services table and calculates the count of each service type.



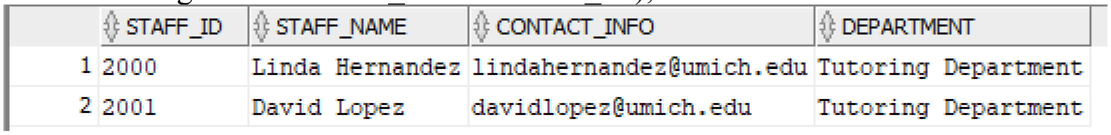
	SERVICE_TYPE	SERVICE_COUNT
1	Online Tutoring	1
2	In Person Tutoring	1
3	Study Hall	1
4	Computer Lab	1

HAVING SQL Statement: SELECT Student_ID, COUNT(*) AS Rental_Count FROM Rentals GROUP BY Student_ID;
Purpose: This query will display all students and their respective rental counts.



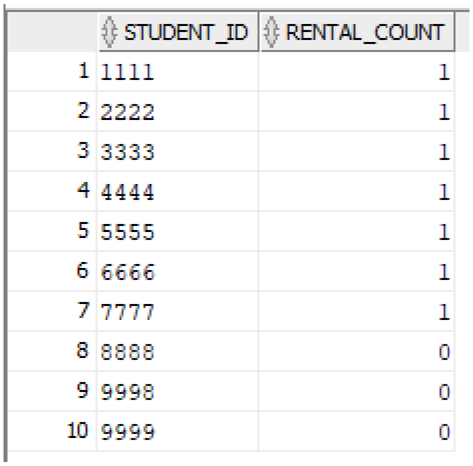
	STUDENT_ID	RENTAL_COUNT
1	2222	1
2	6666	1
3	5555	1
4	1111	1
5	(null)	3
6	4444	1
7	7777	1
8	3333	1

Subqueries
Purpose: This query lists staff members involved in providing tutoring services.
SQL statement: SELECT * FROM Staff_Info WHERE Staff_ID IN (SELECT DISTINCT s.Staff_ID FROM Services s
JOIN Tutoring t ON s.Service_ID = t.Service_ID);



	STAFF_ID	STAFF_NAME	CONTACT_INFO	DEPARTMENT
1	2000	Linda Hernandez	lindahernandez@umich.edu	Tutoring Department
2	2001	David Lopez	davidlopez@umich.edu	Tutoring Department

SQL Statement: SELECT Student_ID, (SELECT COUNT(*) FROM Rentals R WHERE R.Student_ID = S.Student_ID) AS Rental_Count
FROM Student S;
Purpose: The purpose of this query is to retrieve the Student_ID and Name of students who have rentals.



	STUDENT_ID	RENTAL_COUNT
1	1111	1
2	2222	1
3	3333	1
4	4444	1
5	5555	1
6	6666	1
7	7777	1
8	8888	0
9	9998	0
10	9999	0

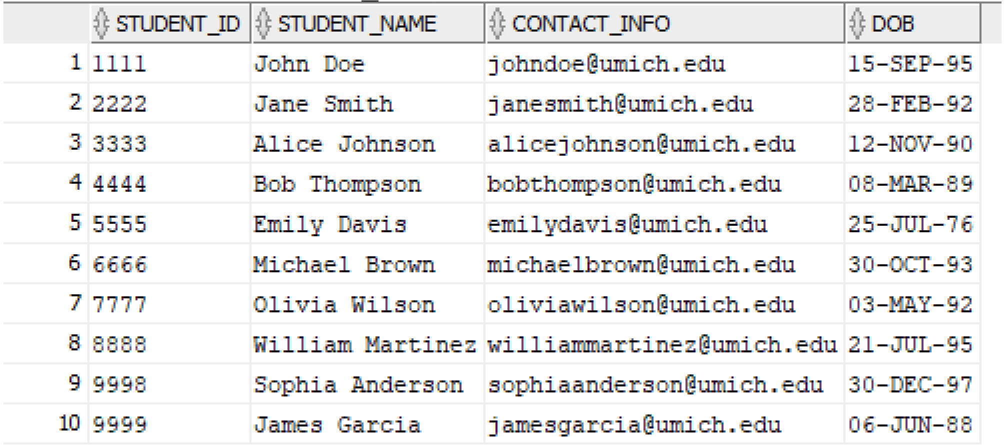
Nested Functions:

.

Documentation of Views

-- View #1 Students_info View
Purpose: This view provides information about students by joining the Student and Person tables.
SQL: CREATE VIEW Students_Info AS
SELECT s.Student_ID, p.Name AS Student_Name, p.Contact_Info, s.DOB FROM Student s
JOIN Person p ON s.Person_ID = p.Person_ID;

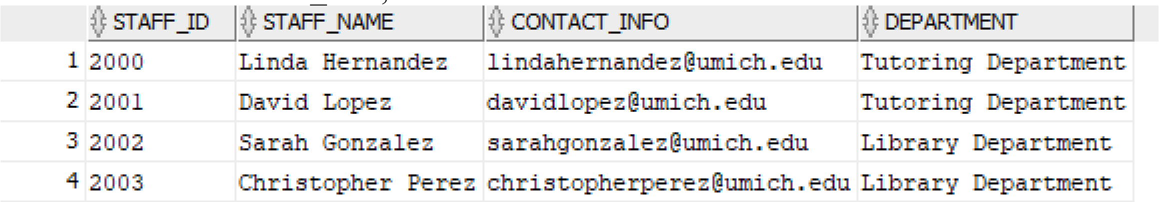
SELECT * FROM Students_Info;



	STUDENT_ID	STUDENT_NAME	CONTACT_INFO	DOB
1	1111	John Doe	johndoe@umich.edu	15-SEP-95
2	2222	Jane Smith	janesmith@umich.edu	28-FEB-92
3	3333	Alice Johnson	alicejohnson@umich.edu	12-NOV-90
4	4444	Bob Thompson	bobthompson@umich.edu	08-MAR-89
5	5555	Emily Davis	emilydavis@umich.edu	25-JUL-76
6	6666	Michael Brown	michaelbrown@umich.edu	30-OCT-93
7	7777	Olivia Wilson	oliviawilson@umich.edu	03-MAY-92
8	8888	William Martinez	williammartinez@umich.edu	21-JUL-95
9	9998	Sophia Anderson	sophiaanderson@umich.edu	30-DEC-97
10	9999	James Garcia	jamesgarcia@umich.edu	06-JUN-88

-- View #2 Staff_Info View
Purpose: This view offers details about staff members by joining the Staff and Person tables.
SQL: CREATE VIEW Staff_Info AS
SELECT st.Staff_ID, p.Name AS Staff_Name, p.Contact_Info, st.Department FROM Staff st
JOIN Person p ON st.Person_ID = p.Person_ID;

SELECT * FROM Staff_Info;

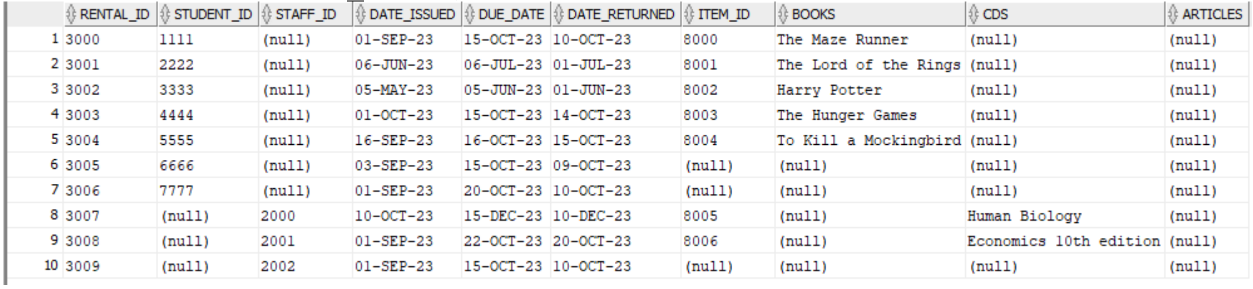


	STAFF_ID	STAFF_NAME	CONTACT_INFO	DEPARTMENT
1	2000	Linda Hernandez	lindahernandez@umich.edu	Tutoring Department
2	2001	David Lopez	davidlopez@umich.edu	Tutoring Department
3	2002	Sarah Gonzalez	sarahgonzalez@umich.edu	Library Department
4	2003	Christopher Perez	christopherperez@umich.edu	Library Department

--View #3 Renatals_Details View
Purpose: This view combines information from the Rentals and Items tables to give an overview of rental transactions.

SQL: CREATE VIEW Rentals_Details AS
SELECT r.Rental_ID, r.Student_ID, r.Staff_ID, r.Date_Issued, r.Due_Date, r.Date_Returned, i.Item_ID, i.Books, i.CDs, i.Articles FROM Rentals r
LEFT JOIN Items i ON r.Rental_ID = i.Rental_ID;

SELECT * FROM Rentals_Details;

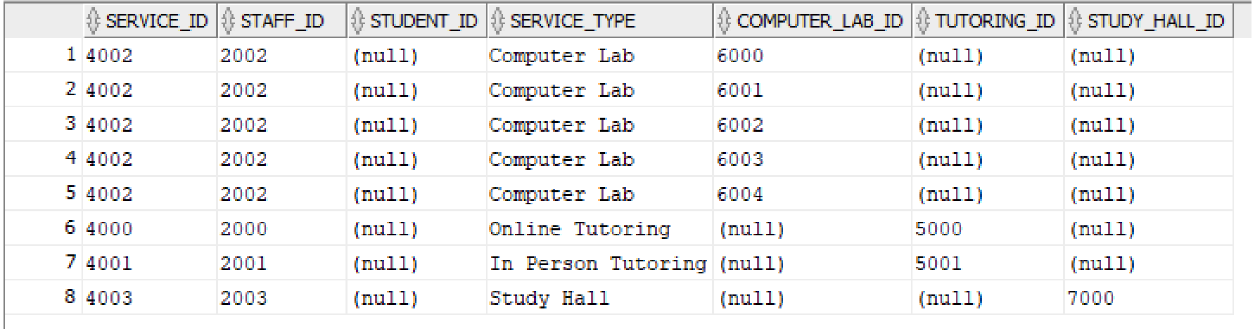


	RENTAL_ID	STUDENT_ID	STAFF_ID	DATE_ISSUED	DUE_DATE	DATE_RETURNED	ITEM_ID	BOOKS	CDS	ARTICLES
1	3000	1111	(null)	01-SEP-23	15-OCT-23	10-OCT-23	8000	The Maze Runner	(null)	(null)
2	3001	2222	(null)	06-JUN-23	06-JUL-23	01-JUL-23	8001	The Lord of the Rings	(null)	(null)
3	3002	3333	(null)	05-MAY-23	05-JUN-23	01-JUN-23	8002	Harry Potter	(null)	(null)
4	3003	4444	(null)	01-OCT-23	15-OCT-23	14-OCT-23	8003	The Hunger Games	(null)	(null)
5	3004	5555	(null)	16-SEP-23	16-OCT-23	15-OCT-23	8004	To Kill a Mockingbird	(null)	(null)
6	3005	6666	(null)	03-SEP-23	15-OCT-23	09-OCT-23	(null)	(null)	(null)	(null)
7	3006	7777	(null)	01-SEP-23	20-OCT-23	10-OCT-23	(null)	(null)	(null)	(null)
8	3007	(null)	2000	10-OCT-23	15-DEC-23	10-DEC-23	8005	(null)	Human Biology	(null)
9	3008	(null)	2001	01-SEP-23	22-OCT-23	20-OCT-23	8006	(null)	Economics 10th edition	(null)
10	3009	(null)	2002	01-SEP-23	15-OCT-23	10-OCT-23	(null)	(null)	(null)	(null)

--View #4 Services_Details View
Purpose: This view consolidates data from the Services, Computer_Lab, Tutoring, and Study_Hall tables to present service-related information. All null values mean that service is not being used at the moment.

SQL: CREATE VIEW Services_Details AS
SELECT s.Service_ID, s.Staff_ID, s.Student_ID, s.Service_Type,
cl.Computer_ID AS Computer_Lab_ID, t.Tutoring_ID AS Tutoring_ID, sh.Study_Hall_ID AS Study_Hall_ID
FROM Services s

```
LEFT JOIN Computer_Lab cl ON s.Service_ID = cl.Service_ID
LEFT JOIN Tutoring t ON s.Service_ID = t.Service_ID
LEFT JOIN Study_Hall sh ON s.Service_ID = sh.Service_ID;

SELECT * FROM Services_Details;
```



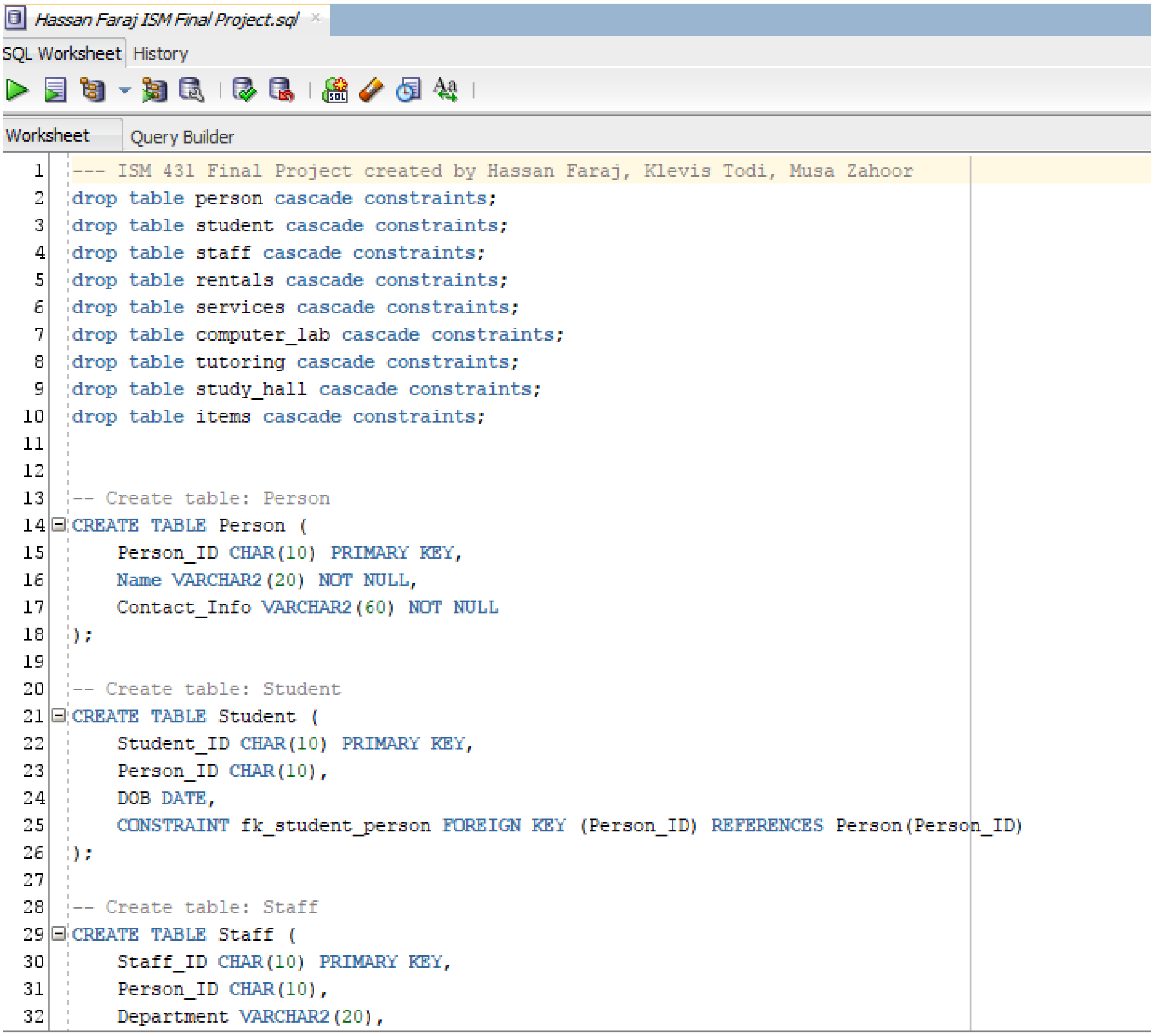
	SERVICE_ID	STAFF_ID	STUDENT_ID	SERVICE_TYPE	COMPUTER_LAB_ID	TUTORING_ID	STUDY_HALL_ID
1	4002	2002	(null)	Computer Lab	6000	(null)	(null)
2	4002	2002	(null)	Computer Lab	6001	(null)	(null)
3	4002	2002	(null)	Computer Lab	6002	(null)	(null)
4	4002	2002	(null)	Computer Lab	6003	(null)	(null)
5	4002	2002	(null)	Computer Lab	6004	(null)	(null)
6	4000	2000	(null)	Online Tutoring	(null)	5000	(null)
7	4001	2001	(null)	In Person Tutoring	(null)	5001	(null)
8	4003	2003	(null)	Study Hall	(null)	(null)	7000

Appendix

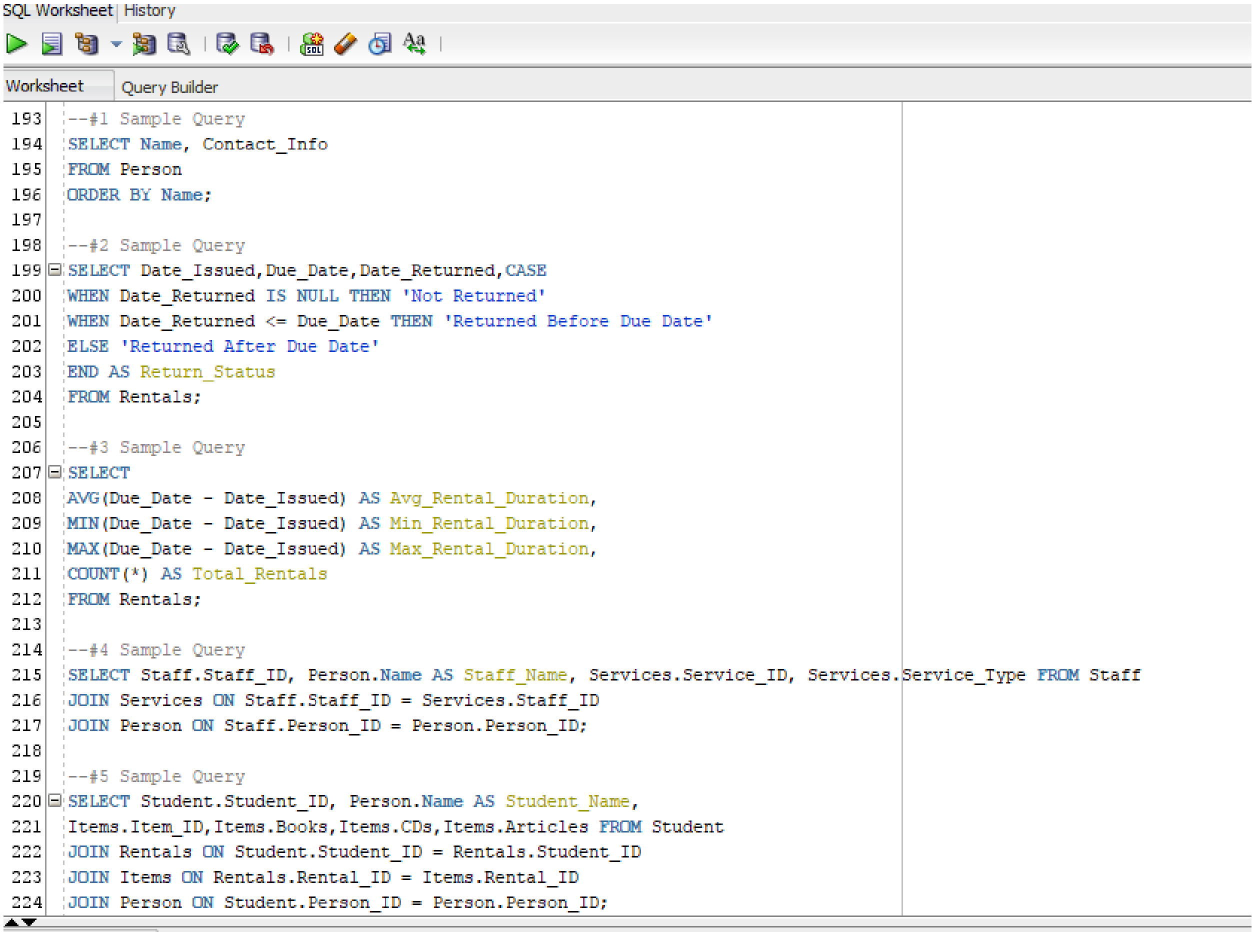
```
/*
Names: Hassan Faraj, Klevis Todi, Musa Zahoor
Course: ISM 431 001 Database Systems II
Professor: Maggie Guo
Date: 11 November 2023
*/
---tables
/*
drop table person cascade constraints;
drop table student cascade constraints;
drop table staff cascade constraints;
drop table rentals cascade constraints;
drop table services cascade constraints;
drop table computer_lab cascade constraints;
drop table tutoring cascade constraints;
drop table study_hall cascade constraints;
drop table items cascade constraints;
*/

-- Create table: Person
CREATE TABLE Person (
  Person_ID CHAR(10) PRIMARY KEY,
  Name VARCHAR2(20) NOT NULL,
  Contact_Info VARCHAR2(60) NOT NULL
```

```
);

-- Create table: Student
CREATE TABLE Student (
   Student_ID CHAR(10) PRIMARY KEY,
   Person_ID CHAR(10),
   DOB DATE,
   CONSTRAINT fk_student_person FOREIGN KEY (Person_ID) REFERENCES Person(Person_ID)
);

-- Create table: Staff
CREATE TABLE Staff (
   Staff_ID CHAR(10) PRIMARY KEY,
   Person_ID CHAR(10),
   Department VARCHAR2(20),
   CONSTRAINT fk_staff_person FOREIGN KEY (Person_ID) REFERENCES Person(Person_ID)
);

-- Create table: Rentals
CREATE TABLE Rentals (
   Rental_ID CHAR(10) PRIMARY KEY,
   Student_ID CHAR(10),
   Staff_ID CHAR(10),
   Date_Issued DATE,
   Due_Date DATE,
   Date_Returned DATE,
    CONSTRAINT fk_rentals_student FOREIGN KEY (Student_ID) REFERENCES Student(Student_ID),
    CONSTRAINT fk_rentals_staff FOREIGN KEY (Staff_ID) REFERENCES Staff(Staff_ID)
);

-- Create table: Services
CREATE TABLE Services (
    Service_ID CHAR(10) PRIMARY KEY,
    Staff_ID CHAR(10),
    Student_ID CHAR(10),
    Service_Type VARCHAR2(200),
    CONSTRAINT fk_services_staff FOREIGN KEY (Staff_ID) REFERENCES Staff(Staff_ID),
    CONSTRAINT fk_services_student FOREIGN KEY (Student_ID) REFERENCES Student(Student_ID)
);

-- Create table: Computer Lab
CREATE TABLE Computer_Lab (
    Computer_ID CHAR(10) PRIMARY KEY,
    Service_ID CHAR(10),
    CONSTRAINT fk_computer_lab_services FOREIGN KEY (Service_ID) REFERENCES Services(Service_ID)
);

-- Create table: Tutoring
CREATE TABLE Tutoring (
    Tutoring_ID CHAR(10) PRIMARY KEY,
    Service_ID CHAR(10),
    CONSTRAINT fk_tutoring_services FOREIGN KEY (Service_ID) REFERENCES
Services(Service_ID)
);

-- Create table: Study Hall
CREATE TABLE Study_Hall (
    Study_Hall_ID CHAR(10) PRIMARY KEY,
    Service_ID CHAR(10),
    CONSTRAINT fk_study_hall_services FOREIGN KEY (Service_ID) REFERENCES
Services(Service_ID)
);

-- Create table: Items
CREATE TABLE Items (
    Item_ID CHAR(10) PRIMARY KEY,
    Rental_ID CHAR(10),
    Books VARCHAR2(200),
    CDs VARCHAR2(200),
    Articles VARCHAR2(200),
    CONSTRAINT fk_items_rentals FOREIGN KEY (Rental_ID) REFERENCES Rentals(Rental_ID)
);

---describe the physical design of each table
describe person;
describe student;

describe staff;

describe rentals;

describe services;

describe computer_lab;

describe tutoring;

describe study_hall;

describe items;


-- Inserting values into the Person table

INSERT INTO Person VALUES ('0001', 'John Doe', 'johndoe@umich.edu');

INSERT INTO Person VALUES ('0002', 'Jane Smith', 'janesmith@umich.edu');

INSERT INTO Person VALUES ('0003', 'Alice Johnson', 'alicejohnson@umich.edu');

INSERT INTO Person VALUES ('0004', 'Bob Thompson', 'bobthompson@umich.edu');

INSERT INTO Person VALUES ('0005', 'Emily Davis', 'emilydavis@umich.edu');

INSERT INTO Person VALUES ('0006', 'Michael Brown', 'michaelbrown@umich.edu');

INSERT INTO Person VALUES ('0007', 'Olivia Wilson', 'oliviawilson@umich.edu');

INSERT INTO Person VALUES ('0008', 'William Martinez', 'williammartinez@umich.edu');

INSERT INTO Person VALUES ('0009', 'Sophia Anderson', 'sophiaanderson@umich.edu');

INSERT INTO Person VALUES ('0010', 'James Garcia', 'jamesgarcia@umich.edu');


INSERT INTO Person VALUES ('0011', 'Linda Hernandez', 'lindahernandez@umich.edu');

INSERT INTO Person VALUES ('0012', 'David Lopez', 'davidlopez@umich.edu');

INSERT INTO Person VALUES ('0013', 'Sarah Gonzalez', 'sarahgonzalez@umich.edu');

INSERT INTO Person VALUES ('0014', 'Christopher Perez', 'christopherperez@umich.edu');

INSERT INTO Person VALUES ('0015', 'Mary Turner', 'maryturner@umich.edu');
```

```
-- Inserting values into the Student table
INSERT INTO Student VALUES ('1111', '0001', TO_DATE('September 15, 1995', 'Month DD, YYYY'));
INSERT INTO Student VALUES ('2222', '0002', TO_DATE('February 28, 1992', 'Month DD, YYYY'));
INSERT INTO Student VALUES ('3333', '0003', TO_DATE('November 12, 1990', 'Month DD, YYYY'));
INSERT INTO Student VALUES ('4444', '0004', TO_DATE('March 8, 1989', 'Month DD, YYYY'));
INSERT INTO Student VALUES ('5555', '0005', TO_DATE('July 25, 1976', 'Month DD, YYYY'));
INSERT INTO Student VALUES ('6666', '0006', TO_DATE('October 30, 1993', 'Month DD, YYYY'));
INSERT INTO Student VALUES ('7777', '0007', TO_DATE('May 3, 1992', 'Month DD, YYYY'));
INSERT INTO Student VALUES ('8888', '0008', TO_DATE('July 21, 1995', 'Month DD, YYYY'));
INSERT INTO Student VALUES ('9998', '0009', TO_DATE('December 30, 1997', 'Month DD, YYYY'));
INSERT INTO Student VALUES ('9999', '0010', TO_DATE('June 6, 1988', 'Month DD, YYYY'));

-- Inserting values into the Staff table
INSERT INTO Staff VALUES ('2000', '0011', 'Tutoring Department');
INSERT INTO Staff VALUES ('2001', '0012', 'Tutoring Department');
INSERT INTO Staff VALUES ('2002', '0013', 'Library Department');
INSERT INTO Staff VALUES ('2003', '0014', 'Library Department');
INSERT INTO Staff VALUES ('2004', '0015', 'Counselor Department');

-- Inserting values into the Rentals table
Insert into rentals values('3000','1111',null, '01-sep-23', '15-oct-23', '10-oct-23');
Insert into rentals values('3001','2222',null, '06-jun-23', '06-jul-23', '01-jul-23');
Insert into rentals values('3002','3333',null, '05-may-23', '05-jun-23', '01-jun-23');

Insert into rentals values('3003','4444',null, '01-oct-23', '15-oct-23', '14-oct-23');

Insert into rentals values('3004','5555',null, '16-sep-23', '16-oct-23', '15-oct-23');

Insert into rentals values('3005','6666',null, '03-sep-23', '15-oct-23', '09-oct-23');

Insert into rentals values('3006','7777',null, '01-sep-23', '20-oct-23', '10-oct-23');


Insert into rentals values('3007', null,'2000', '10-oct-23', '15-dec-23', '10-dec-23');

Insert into rentals values('3008', null,'2001', '01-sep-23', '22-oct-23', '20-oct-23');

Insert into rentals values('3009', null,'2002', '01-sep-23', '15-oct-23', '10-oct-23');


-- Inserting values into the services table

insert into services values('4000','2000',null,'Online Tutoring');

insert into services values('4001','2001',null,'In Person Tutoring');

insert into services values('4002','2002',null,'Computer Lab');

insert into services values('4003','2003',null,'Study Hall');


-- Inserting values into Computer_lab table

insert into computer_lab values('6000','4002');

insert into computer_lab values('6001','4002');

insert into computer_lab values('6002','4002');

insert into computer_lab values('6003','4002');

insert into computer_lab values('6004','4002');



-- Inserting values into tutoring table

Insert into tutoring values ('5000','4000');
Insert into tutoring values ('5001','4001');



--Inserting values into Study_Hall table

Insert into study_hall values('7000', '4003');



--Inserting books into items table

insert into items values('8000','3000','The Maze Runner',null,null);

insert into items values('8001','3001','The Lord of the Rings',null,null);

insert into items values('8002','3002','Harry Potter',null,null);

insert into items values('8003','3003','The Hunger Games',null,null);

insert into items values('8004','3004','To Kill a Mockingbird',null,null);


--Inserting CDs into items table

insert into items values('8005','3007',null,'Human Biology',null);

insert into items values('8006','3008',null,'Economics 10th edition',null);


--Inserting articles into items table

insert into items values('8007','3009',null,null,'Racial Injustice');

insert into items values('8008','3005',null,null,'Economics in the world today');

insert into items values('8009','3006',null,null,'Samsung vs Apple');


Commit;
```



```sql
--- ISM 431 Final Project created by Hassan Faraj, Klevis Todi, Musa Zahoor
drop table person cascade constraints;
drop table student cascade constraints;
drop table staff cascade constraints;
drop table rentals cascade constraints;
drop table services cascade constraints;
drop table computer_lab cascade constraints;
drop table tutoring cascade constraints;
drop table study_hall cascade constraints;
drop table items cascade constraints;


-- Create table: Person
CREATE TABLE Person (
    Person_ID CHAR(10) PRIMARY KEY,
    Name VARCHAR2(20) NOT NULL,
    Contact_Info VARCHAR2(60) NOT NULL
);

-- Create table: Student
CREATE TABLE Student (
    Student_ID CHAR(10) PRIMARY KEY,
    Person_ID CHAR(10),
    DOB DATE,
    CONSTRAINT fk_student_person FOREIGN KEY (Person_ID) REFERENCES Person(Person_ID)
);

-- Create table: Staff
CREATE TABLE Staff (
    Staff_ID CHAR(10) PRIMARY KEY,
    Person_ID CHAR(10),
    Department VARCHAR2(20),
```



```sql
--#1 Sample Query
SELECT Name, Contact_Info
FROM Person
ORDER BY Name;

--#2 Sample Query
SELECT Date_Issued,Due_Date,Date_Returned,CASE
WHEN Date_Returned IS NULL THEN 'Not Returned'
WHEN Date_Returned <= Due_Date THEN 'Returned Before Due Date'
ELSE 'Returned After Due Date'
END AS Return_Status
FROM Rentals;

--#3 Sample Query
SELECT
AVG(Due_Date - Date_Issued) AS Avg_Rental_Duration,
MIN(Due_Date - Date_Issued) AS Min_Rental_Duration,
MAX(Due_Date - Date_Issued) AS Max_Rental_Duration,
COUNT(*) AS Total_Rentals
FROM Rentals;

--#4 Sample Query
SELECT Staff.Staff_ID, Person.Name AS Staff_Name, Services.Service_ID, Services.Service_Type FROM Staff
JOIN Services ON Staff.Staff_ID = Services.Staff_ID
JOIN Person ON Staff.Person_ID = Person.Person_ID;

--#5 Sample Query
SELECT Student.Student_ID, Person.Name AS Student_Name,
Items.Item_ID,Items.Books,Items.CDs,Items.Articles FROM Student
JOIN Rentals ON Student.Student_ID = Rentals.Student_ID
JOIN Items ON Rentals.Rental_ID = Items.Rental_ID
JOIN Person ON Student.Person_ID = Person.Person_ID;
```