Chrissie Raj Bezzam
G01465544

1) Practice Problems: Data Manipulation and Transaction Control

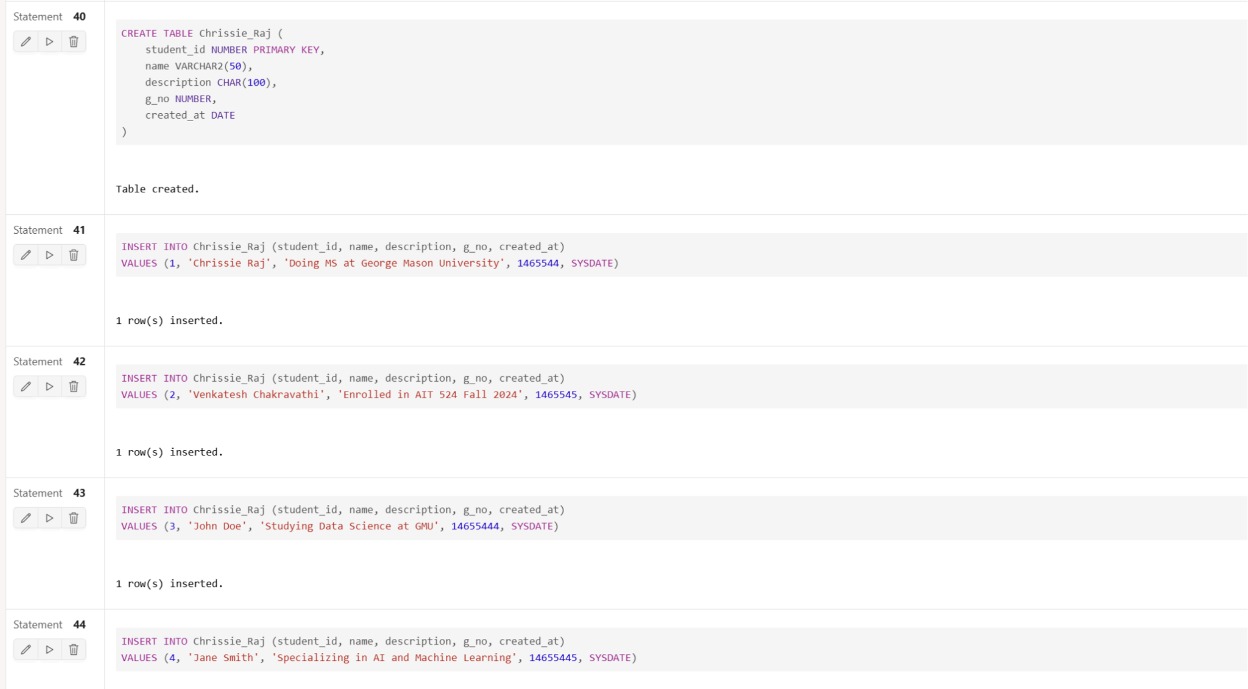
- **Create a new table containing at least four columns of four different data types (CHAR, VARCHAR2, NUMBER, DATE). Name the table your_first_name_your_last_name (substitute your_first_name with your first name and your_last_name with your last name).**
- **Insert a new record into the table created in Problem 1. When inserting a record, provide values for all columns. Use the default format for the date. Use the SELECT * FROM your_table_name; command to display the content of the table (make sure to substitute your_table_name with the actual name of your table).**
- **Insert a new record into the table created in Problem 1. When inserting a record, provide values for selected columns only. Use the default format for the date. Use the SELECT * FROM your_table_name; command to display the content of the table (make sure to substitute your_table_name with the actual name of your table).**
- **Update multiple values in the table created in Problem 1. Use the SELECT * FROM your_table_name; command to display the content of the table (make sure to substitute your_table_name with the actual name of your table).**
- **Delete one specific row from the table created in Problem 1. Use the SELECT * FROM your_table_name; command to display the content of the table (make sure to substitute your_table_name with the actual name of your table).**
- **Use the SELECT * FROM your_table_name; command to display the content of the table (make sure to substitute your_table_name with the actual name of your table). Delete one specific row from the table created in Problem 1. Use the SELECT * FROM your_table_name; command to display the content of the table (make sure to substitute your_table_name with the actual name of your table). Execute a command that undoes the deletion. Use the SELECT * FROM your_table_name; command to display the content of the table (make sure to substitute your_table_name with the actual name of your table). Note: In Oracle Live , you need to execute all the commands together as a single transaction to make ROLLBACK work. Since Oracle Live auto commits transactions, you can't rollback after a command is executed (for instance, you can't first delete a record and then**

use ROLLBACK to reverse the changes after DELETE was executed). To see how ROLLBACK works in Oracle Live , you need to include ROLLBACK within your transaction as suggested above.

- **Save the changes permanently to the database.**
- **Create a script using substitution variables that allows a user to set a new value for one of the values in the table created in Problem 1 based on its PK value. Skip this problem if you use Oracle Live (it does not support substitution variables). Note: Some of the features and commands that we are learning in class might not be fully supported in Oracle Live. If you use Plus, substitution variables should work. If you use Oracle Live , either answer this question by following the studied concepts without testing it, or skip it.**
- **In the table created in Problem 1, find a column that contains numerical values and perform a meaningful arithmetic operation on data. If there is no such column, add a new column. Explain in a complete, coherent sentence what the query is intended to do.**
- **Delete the table created in Problem 1.**

SOLUTION:

SCREENSHOTS:



Statement 40

```
CREATE TABLE Chrissie_Raj (
    student_id NUMBER PRIMARY KEY,
    name VARCHAR2(50),
    description CHAR(100),
    g_no NUMBER,
    created_at DATE
)
```

```
Table created.
```

Statement 41

```
INSERT INTO Chrissie_Raj (student_id, name, description, g_no, created_at)
VALUES (1, 'Chrissie Raj', 'Doing MS at George Mason University', 1465544, SYSDATE)
```

```
1 row(s) inserted.
```

Statement 42

```
INSERT INTO Chrissie_Raj (student_id, name, description, g_no, created_at)
VALUES (2, 'Venkatesh Chakravathi', 'Enrolled in AIT 524 Fall 2024', 1465545, SYSDATE)
```

```
1 row(s) inserted.
```

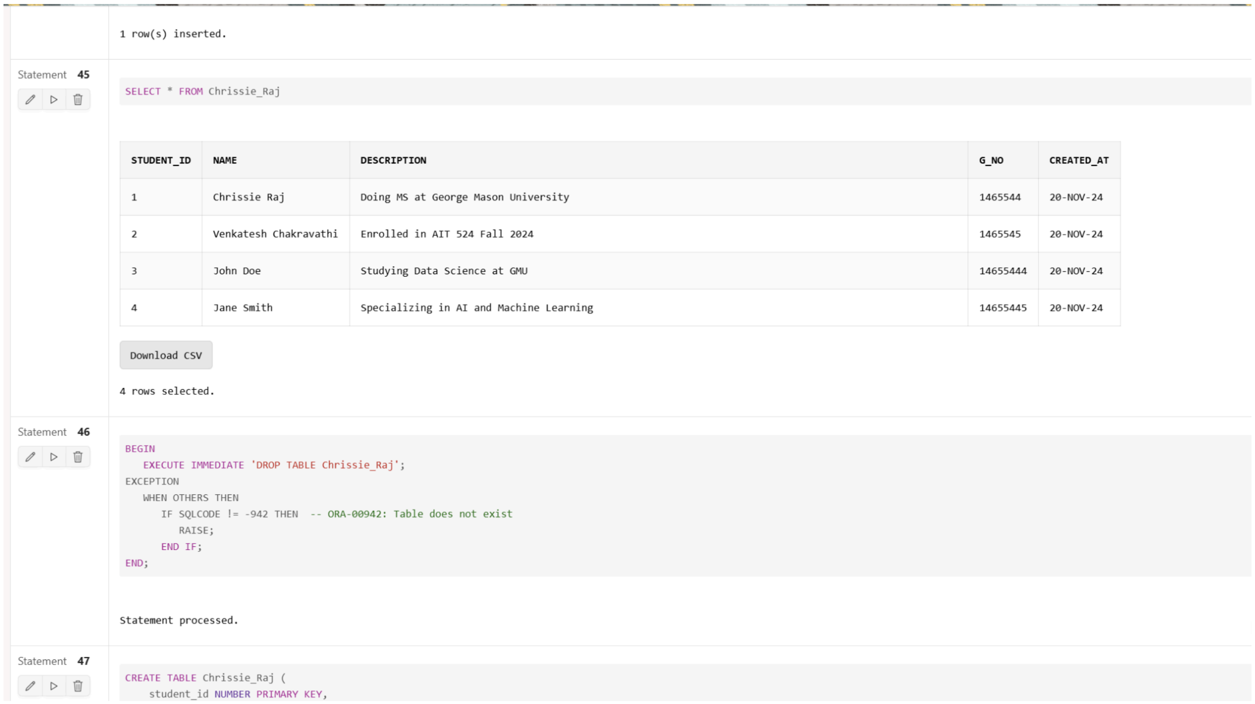
Statement 43

```
INSERT INTO Chrissie_Raj (student_id, name, description, g_no, created_at)
VALUES (3, 'John Doe', 'Studying Data Science at GMU', 14655444, SYSDATE)
```

```
1 row(s) inserted.
```

Statement 44

```
INSERT INTO Chrissie_Raj (student_id, name, description, g_no, created_at)
VALUES (4, 'Jane Smith', 'Specializing in AI and Machine Learning', 14655445, SYSDATE)
```



```
1 row(s) inserted.
```

Statement 45

```
SELECT * FROM Chrissie_Raj
```

STUDENT_ID	NAME	DESCRIPTION	G_NO	CREATED_AT
1	Chrissie Raj	Doing MS at George Mason University	1465544	20-NOV-24
2	Venkatesh Chakravathi	Enrolled in AIT 524 Fall 2024	1465545	20-NOV-24
3	John Doe	Studying Data Science at GMU	14655444	20-NOV-24
4	Jane Smith	Specializing in AI and Machine Learning	14655445	20-NOV-24

```
4 rows selected.
```

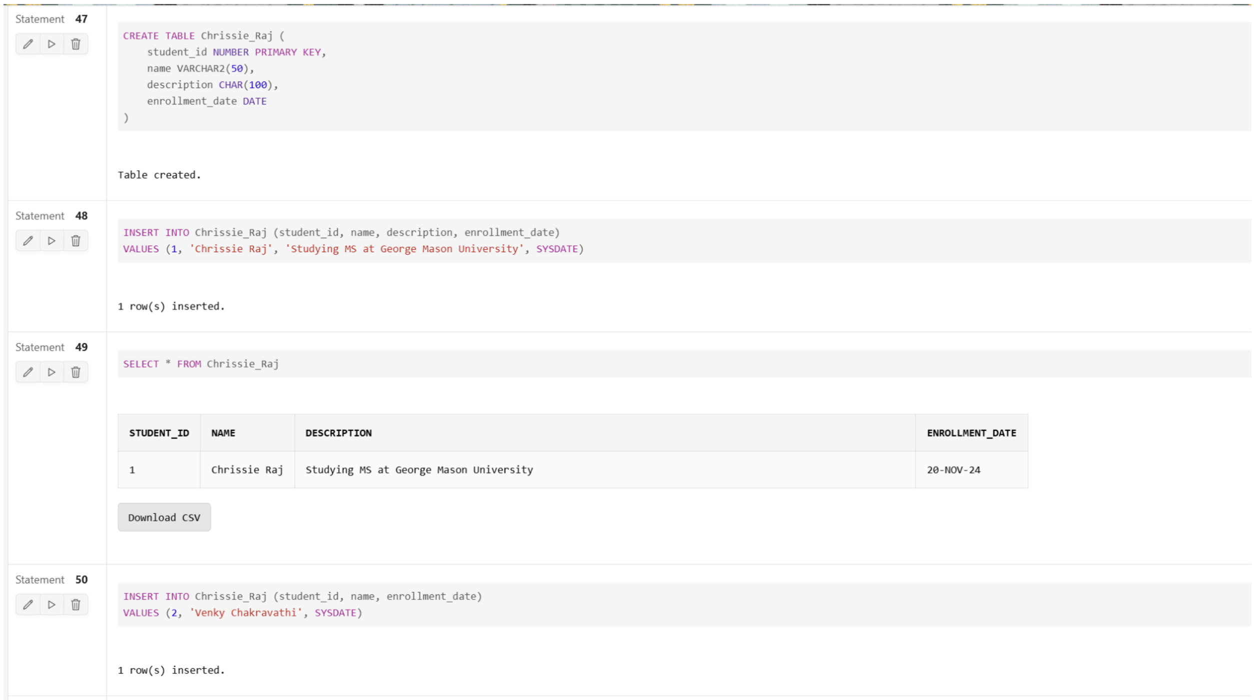
Statement 46

```
BEGIN
   EXECUTE IMMEDIATE 'DROP TABLE Chrissie_Raj';
EXCEPTION
   WHEN OTHERS THEN
      IF SQLCODE != -942 THEN  -- ORA-00942: Table does not exist
         RAISE;
      END IF;
END;
```

```
Statement processed.
```

Statement 47

```
CREATE TABLE Chrissie_Raj (
    student_id NUMBER PRIMARY KEY,
```



Statement 47

```
CREATE TABLE Chrissie_Raj (
    student_id NUMBER PRIMARY KEY,
    name VARCHAR2(50),
    description CHAR(100),
    enrollment_date DATE
)
```

```
Table created.
```

Statement 48

```
INSERT INTO Chrissie_Raj (student_id, name, description, enrollment_date)
VALUES (1, 'Chrissie Raj', 'Studying MS at George Mason University', SYSDATE)
```

```
1 row(s) inserted.
```

Statement 49

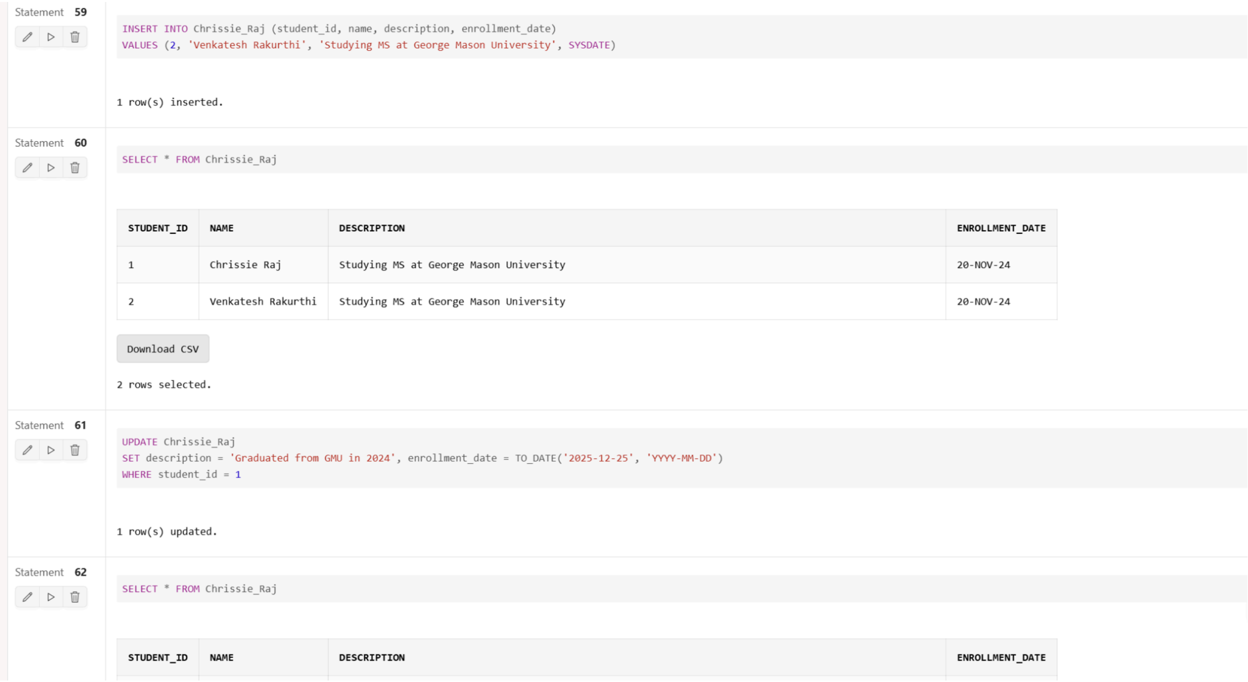
```
SELECT * FROM Chrissie_Raj
```

STUDENT_ID	NAME	DESCRIPTION	ENROLLMENT_DATE
1	Chrissie Raj	Studying MS at George Mason University	20-NOV-24

Statement 50

```
INSERT INTO Chrissie_Raj (student_id, name, enrollment_date)
VALUES (2, 'Venky Chakravathi', SYSDATE)
```

```
1 row(s) inserted.
```



Statement 59

```
INSERT INTO Chrissie_Raj (student_id, name, description, enrollment_date)
VALUES (2, 'Venkatesh Rakurthi', 'Studying MS at George Mason University', SYSDATE)
```

```
1 row(s) inserted.
```

Statement 60

```
SELECT * FROM Chrissie_Raj
```

STUDENT_ID	NAME	DESCRIPTION	ENROLLMENT_DATE
1	Chrissie Raj	Studying MS at George Mason University	20-NOV-24
2	Venkatesh Rakurthi	Studying MS at George Mason University	20-NOV-24

```
2 rows selected.
```

Statement 61

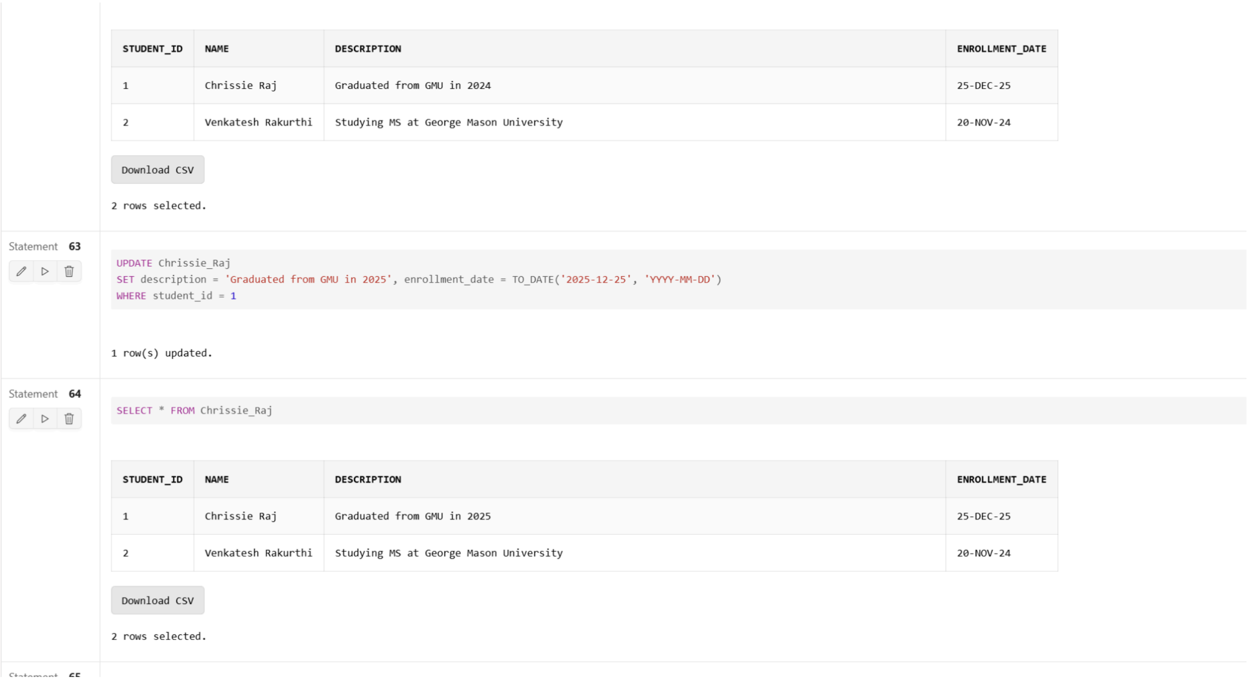
```
UPDATE Chrissie_Raj
SET description = 'Graduated from GMU in 2024', enrollment_date = TO_DATE('2025-12-25', 'YYYY-MM-DD')
WHERE student_id = 1
```

```
1 row(s) updated.
```

Statement 62

```
SELECT * FROM Chrissie_Raj
```

STUDENT_ID	NAME	DESCRIPTION	ENROLLMENT_DATE



STUDENT_ID	NAME	DESCRIPTION	ENROLLMENT_DATE
1	Chrissie Raj	Graduated from GMU in 2024	25-DEC-25
2	Venkatesh Rakurthi	Studying MS at George Mason University	20-NOV-24

```
2 rows selected.
```

Statement 63

```sql
UPDATE Chrissie_Raj
SET description = 'Graduated from GMU in 2025', enrollment_date = TO_DATE('2025-12-25', 'YYYY-MM-DD')
WHERE student_id = 1
```

```
1 row(s) updated.
```

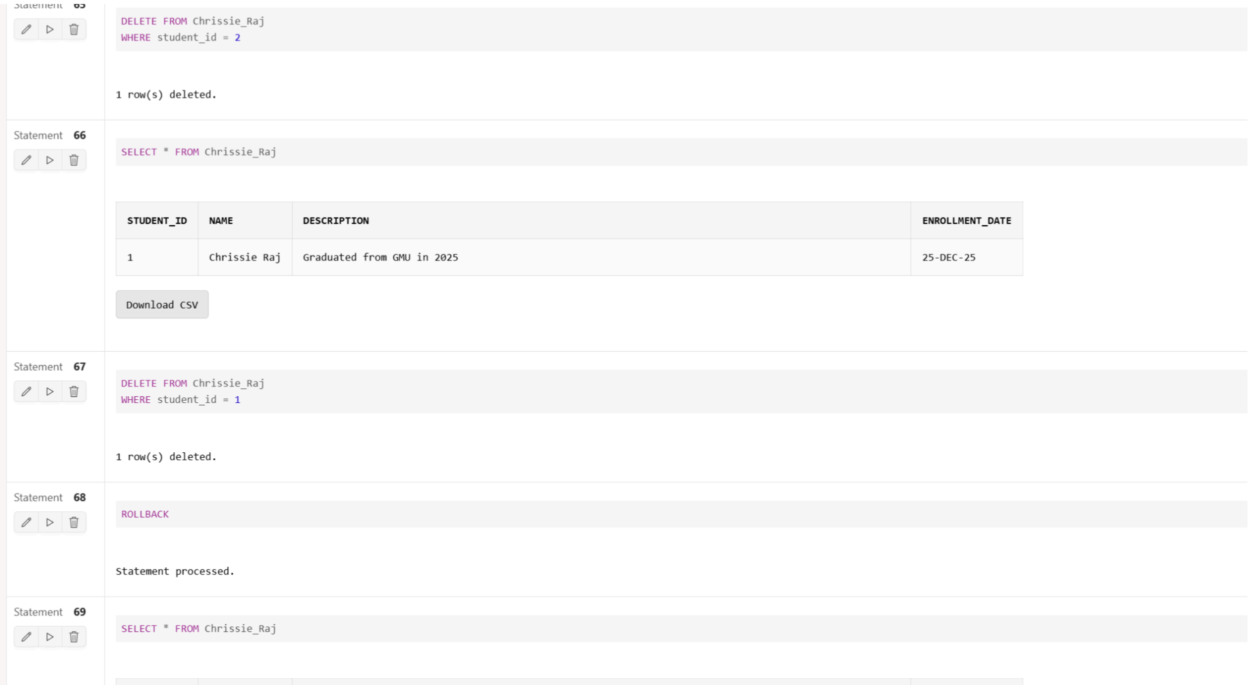
Statement 64

```sql
SELECT * FROM Chrissie_Raj
```

STUDENT_ID	NAME	DESCRIPTION	ENROLLMENT_DATE
1	Chrissie Raj	Graduated from GMU in 2025	25-DEC-25
2	Venkatesh Rakurthi	Studying MS at George Mason University	20-NOV-24

```
2 rows selected.
```

Statement 65



Statement 65

```
DELETE FROM Chrissie_Raj
WHERE student_id = 2
```

```
1 row(s) deleted.
```

Statement 66

```
SELECT * FROM Chrissie_Raj
```

STUDENT_ID	NAME	DESCRIPTION	ENROLLMENT_DATE
1	Chrissie Raj	Graduated from GMU in 2025	25-DEC-25

Statement 67

```
DELETE FROM Chrissie_Raj
WHERE student_id = 1
```

```
1 row(s) deleted.
```

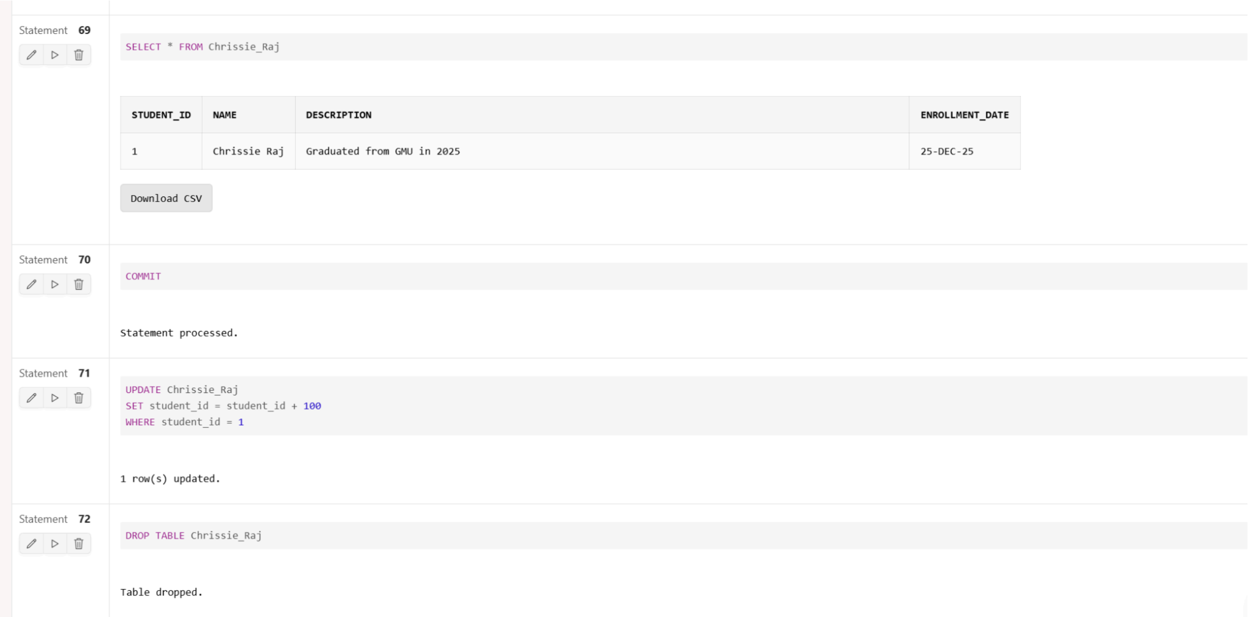
Statement 68

```
ROLLBACK
```

```
Statement processed.
```

Statement 69

```
SELECT * FROM Chrissie_Raj
```



Statement 69

```
SELECT * FROM Chrissie_Raj
```

STUDENT_ID	NAME	DESCRIPTION	ENROLLMENT_DATE
1	Chrissie Raj	Graduated from GMU in 2025	25-DEC-25

Statement 70

```
COMMIT
```

```
Statement processed.
```

Statement 71

```
UPDATE Chrissie_Raj
SET student_id = student_id + 100
WHERE student_id = 1
```

```
1 row(s) updated.
```

Statement 72

```
DROP TABLE Chrissie_Raj
```

```
Table dropped.
```

1. Create a New Table

The first problem requires creating a new table named after your first and last name, which should contain at least four columns of different data types: `CHAR`, `VARCHAR2`, `NUMBER`, and `DATE`. This task helps familiarize you with different data types available in and how to define columns using these types to store different kinds of information. Creating a table is foundational for any database work since it establishes where the data will be stored.

Table Schema:

```
CREATE TABLE Chrissie_Raj (
    student_id NUMBER PRIMARY KEY,
    name VARCHAR2(50),
    description CHAR(100),
    enrollment_date DATE
);
```

This schema includes a `student_id` (primary key), a `name` column for storing variable-length names, a `description` column for a fixed-length description, and `enrollment_date` to store the enrollment date of a student.

---

2. Insert a Record with All Columns

The next task involves inserting a record into the newly created table, providing values for all columns. This teaches how to insert data completely into each field of a table.

:

```
INSERT INTO Chrissie_Raj (student_id, name, description,
enrollment_date)
VALUES (1, 'Chrissie Raj', 'Doing MS at George Mason
University', SYSDATE);
```

This shows how to insert all values, including the current system date (`SYSDATE`) for the enrollment date.

---

3. Insert a Record with Selected Columns

This task involves inserting a new record by providing values for only selected columns, demonstrating flexibility in data insertion.

:

```
INSERT INTO Chrissie_Raj (student_id, name, enrollment_date)
VALUES (2, 'Venkatesh Rakurthi', SYSDATE);
```

Here, only selected fields (`student_id`, `name`, and `enrollment_date`) are filled, leaving others (like `description`) as `NULL`.

---

4. Update Multiple Values

The next step is updating multiple columns in the table, which demonstrates how to modify existing data.

:

```
UPDATE Chrissie_Raj
SET description = 'Graduated from GMU in 2024',
enrollment_date = TO_DATE('2024-06-15', 'YYYY-MM-DD')
WHERE student_id = 1;
```

This command changes both the `description` and the `enrollment_date` for the record where `student_id` is 1.

---

5. Delete One Specific Row

This task involves deleting a specific row from the table. This teaches how to remove unwanted data records from a table.

:

```
DELETE FROM Chrissie_Raj
WHERE student_id = 2;

SELECT * FROM Chrissie_Raj;
```

The `DELETE` statement is used to remove the record where `student_id` is 2, and the `SELECT` statement is used to verify the current content of the table.

---

6. Delete a Row and Use ROLLBACK

This part of the assignment involves deleting a specific row and then rolling back the deletion, demonstrating transaction control in Oracle. However, Oracle Live automatically commits changes, meaning `ROLLBACK` can only work if all commands are executed together in a single transaction.

:

```
DELETE FROM Chrissie_Raj
WHERE student_id = 1;

ROLLBACK;

SELECT * FROM Chrissie_Raj;
```

---

7. Save Changes Permanently

This requires saving changes to the database permanently using the `COMMIT` command. It ensures that all previous changes, such as updates and deletions, are saved:

```
COMMIT;
```

This command finalizes all modifications, making them permanent in the database.

---

8. Script with Substitution Variables

This part asks to create a script using substitution variables to update a value in the table based on its primary key (`PK`). Oracle Live doesn't support substitution variables, so this problem is only applicable if using Plus.

:

```
UPDATE Chrissie_Raj
SET description = '&new_description'
```

```
WHERE student_id = &student_id;
```

This allows a user to provide a new description for a specific `student_id`.

---

9. Arithmetic Operation on a Numeric Column

This task asks for performing an arithmetic operation on a numerical column, such as incrementing a value. If there is no numerical column available, one should be added.

:

```
UPDATE Chrissie_Raj
SET student_id = student_id + 100
WHERE student_id = 1;

-- This query increases the `student_id` by 100 for the
student with `student_id = 1`.
```

---

10. Delete the Table

Finally, the task asks to delete the table created in Problem 1. This demonstrates the ability to completely remove a table and its data from the database.

:

```
DROP TABLE Chrissie_Raj;
```

This command removes the entire table and all the data within it.

2)Practice Problems: Restricting Rows and Sorting Data

Before starting these problems, update the JustLee Books database by executing the JLDB_Build_Extended.sql script. You can find the script in the "Class Databases" folder.

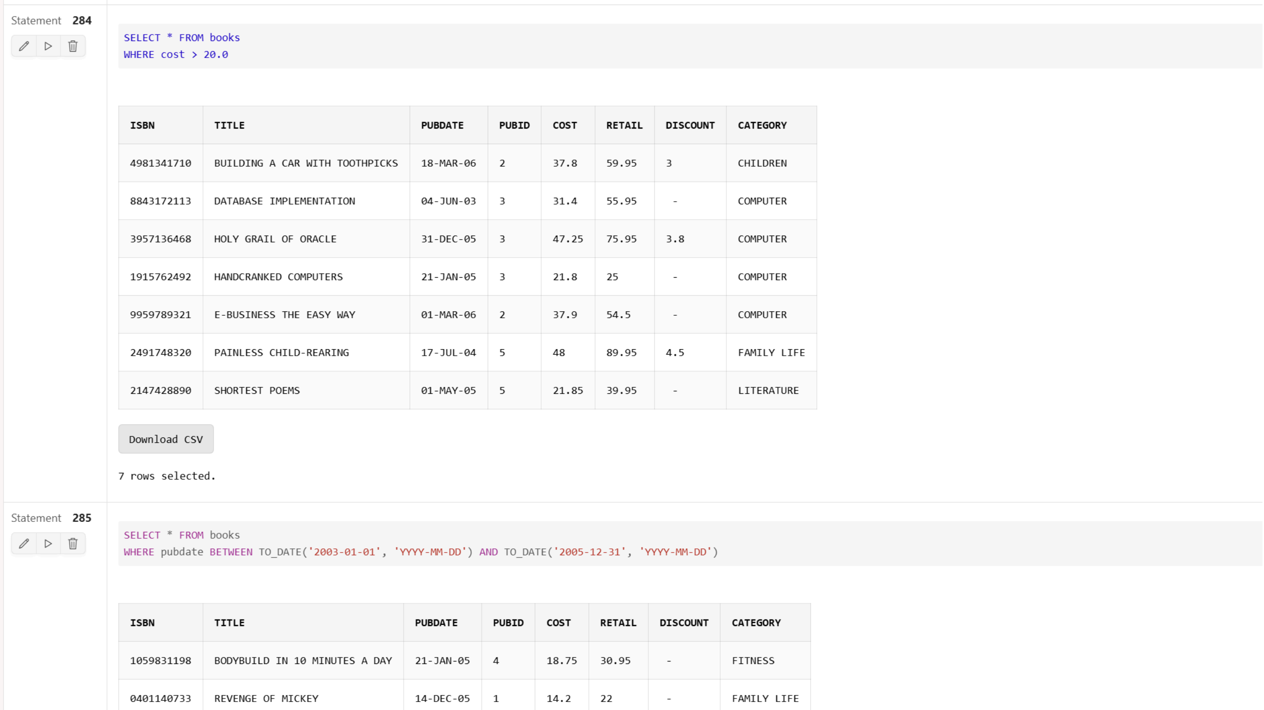
- **Write an SQL query to retrieve records from one of the tables in the JustLee Books database. In a search condition, include one or more arithmetic comparison operators (=, !=, >, <, >=, etc.). Explain in a complete, coherent sentence what the query is intended to do.**
- **Write an SQL query to retrieve records from one of the tables in the JustLee Books database. In a search condition, include the BETWEEN … AND operator. Explain in a complete, coherent sentence what the query is intended to do.**
- **Write an SQL query to retrieve records from one of the tables in the JustLee Books database. In a search condition, include the IN operator. Explain in a complete, coherent sentence what the query is intended to do.**
- **Write an SQL query to retrieve records from one of the tables in the JustLee Books database. In a search condition, include the LIKE operator with either % or _ (or both). Explain in a complete, coherent sentence what the query is intended to do.**
- **Write an SQL query to retrieve records from one of the tables in the JustLee Books database. In a search condition, include the IS NULL operator. Explain in a complete, coherent sentence what the query is intended to do.**
- **Write an SQL query to retrieve records from one of the tables in the JustLee Books database. In a search condition, include multiple conditions using logical operators AND and OR. Explain in a complete, coherent sentence what the query is intended to do.**
- **Write a complex SQL query to retrieve records from one of the tables in the JustLee Books database. In a search condition, use logical operators to join multiple conditions that include at least one of the arithmetic operators and one of the special operators. Use the ORDER BY statement to order the**

output. Explain in a complete, coherent sentence what the query is intended to do.

- List the title and publish date of any computer book published in 2005. Perform the task of searching for the publish date by using one of the three methods: a) a range operator, b) a logical operator, and c) a search pattern operation.
- Write an SQL query to address the following scenario: A manager at JustLee Books requests a list of the titles of all books generating a profit of at least $10.00. The manager wants the results listed in descending order, based on each book's profit.
- Write an SQL query to address the following scenario: A customer service representative is trying to identify all books in the Computer or Family Life category and published by Publisher 1 or Publisher 3. However, the results shouldn't include any book selling for less than $45.00.

Solution:

9 SCREENSHOTS:



Statement 284

```
SELECT * FROM books
WHERE cost > 20.0
```

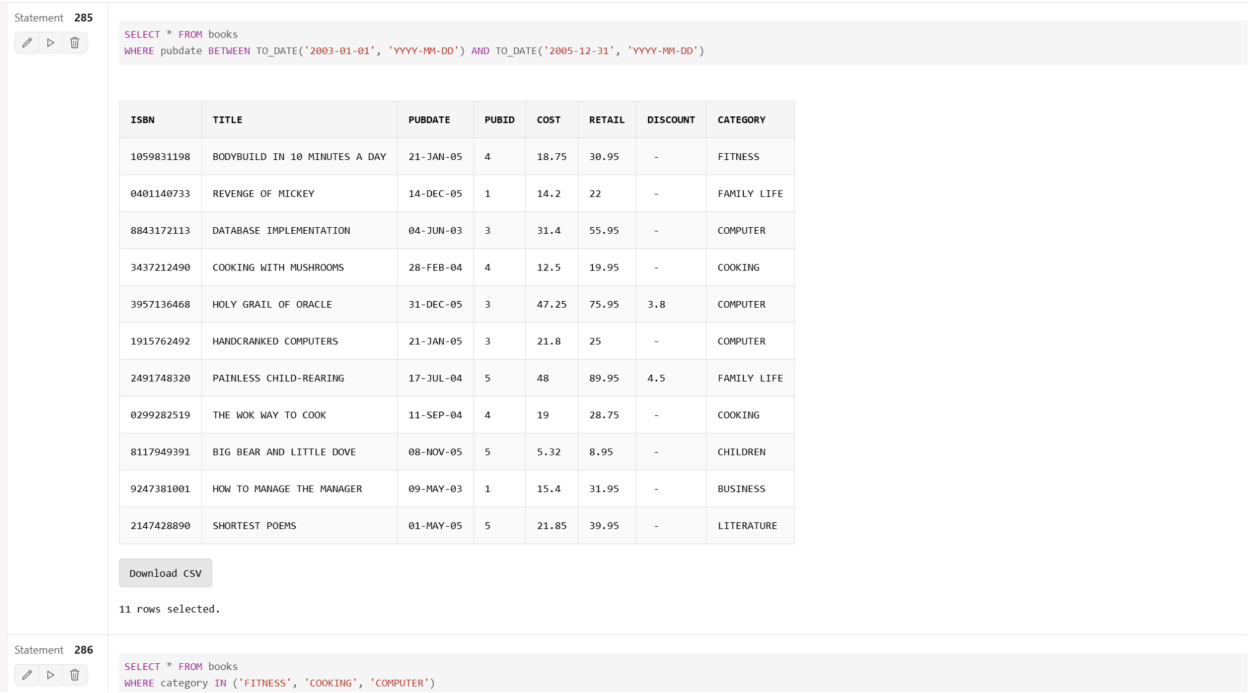
ISBN	TITLE	PUBDATE	PUBID	COST	RETAIL	DISCOUNT	CATEGORY
4981341710	BUILDING A CAR WITH TOOTHPICKS	18-MAR-06	2	37.8	59.95	3	CHILDREN
8843172113	DATABASE IMPLEMENTATION	04-JUN-03	3	31.4	55.95	-	COMPUTER
3957136468	HOLY GRAIL OF ORACLE	31-DEC-05	3	47.25	75.95	3.8	COMPUTER
1915762492	HANDCRANKED COMPUTERS	21-JAN-05	3	21.8	25	-	COMPUTER
9959789321	E-BUSINESS THE EASY WAY	01-MAR-06	2	37.9	54.5	-	COMPUTER
2491748320	PAINLESS CHILD-REARING	17-JUL-04	5	48	89.95	4.5	FAMILY LIFE
2147428890	SHORTEST POEMS	01-MAY-05	5	21.85	39.95	-	LITERATURE

7 rows selected.

Statement 285

```
SELECT * FROM books
WHERE pubdate BETWEEN TO_DATE('2003-01-01', 'YYYY-MM-DD') AND TO_DATE('2005-12-31', 'YYYY-MM-DD')
```

ISBN	TITLE	PUBDATE	PUBID	COST	RETAIL	DISCOUNT	CATEGORY
1059831198	BODYBUILD IN 10 MINUTES A DAY	21-JAN-05	4	18.75	30.95	-	FITNESS
0401140733	REVENGE OF MICKEY	14-DEC-05	1	14.2	22	-	FAMILY LIFE



Statement 285

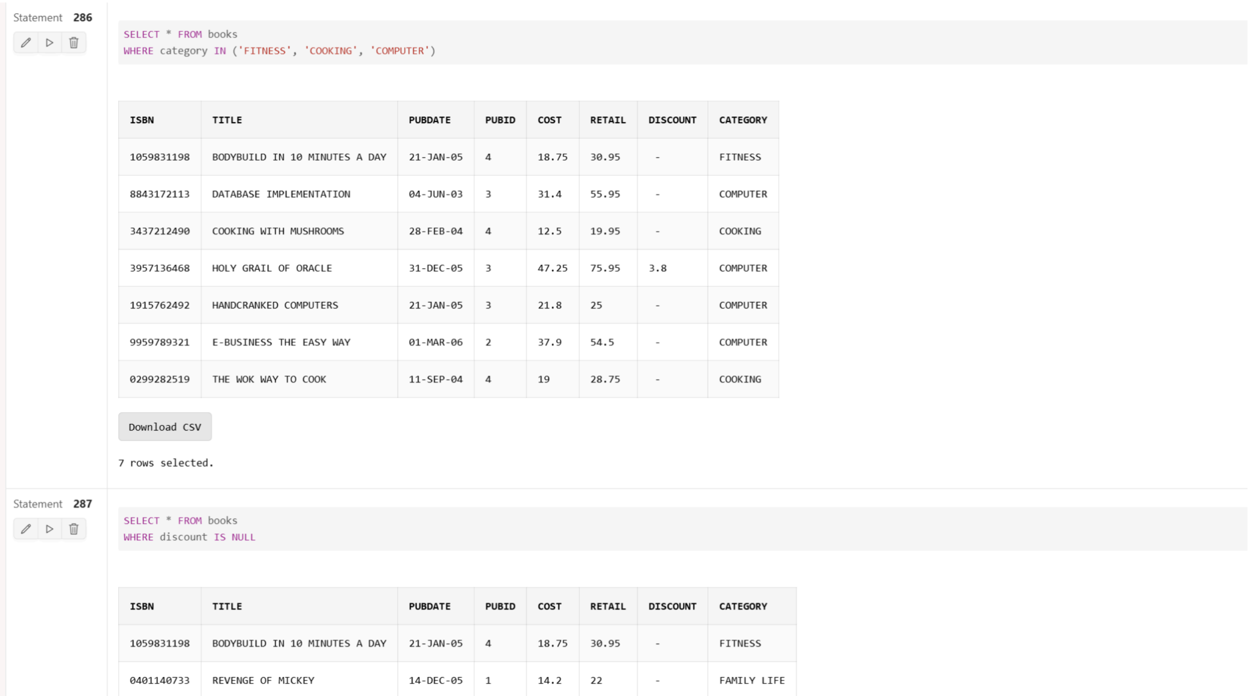
```
SELECT * FROM books
WHERE pubdate BETWEEN TO_DATE('2003-01-01', 'YYYY-MM-DD') AND TO_DATE('2005-12-31', 'YYYY-MM-DD')
```

ISBN	TITLE	PUBDATE	PUBID	COST	RETAIL	DISCOUNT	CATEGORY
1059831198	BODYBUILD IN 10 MINUTES A DAY	21-JAN-05	4	18.75	30.95	-	FITNESS
0401140733	REVENGE OF MICKEY	14-DEC-05	1	14.2	22	-	FAMILY LIFE
8843172113	DATABASE IMPLEMENTATION	04-JUN-03	3	31.4	55.95	-	COMPUTER
3437212490	COOKING WITH MUSHROOMS	28-FEB-04	4	12.5	19.95	-	COOKING
3957136468	HOLY GRAIL OF ORACLE	31-DEC-05	3	47.25	75.95	3.8	COMPUTER
1915762492	HANDCRANKED COMPUTERS	21-JAN-05	3	21.8	25	-	COMPUTER
2491748320	PAINLESS CHILD-REARING	17-JUL-04	5	48	89.95	4.5	FAMILY LIFE
0299282519	THE WOK WAY TO COOK	11-SEP-04	4	19	28.75	-	COOKING
8117949391	BIG BEAR AND LITTLE DOVE	08-NOV-05	5	5.32	8.95	-	CHILDREN
9247381001	HOW TO MANAGE THE MANAGER	09-MAY-03	1	15.4	31.95	-	BUSINESS
2147428890	SHORTEST POEMS	01-MAY-05	5	21.85	39.95	-	LITERATURE

11 rows selected.

Statement 286

```
SELECT * FROM books
WHERE category IN ('FITNESS', 'COOKING', 'COMPUTER')
```



Statement 286

```
SELECT * FROM books
WHERE category IN ('FITNESS', 'COOKING', 'COMPUTER')
```

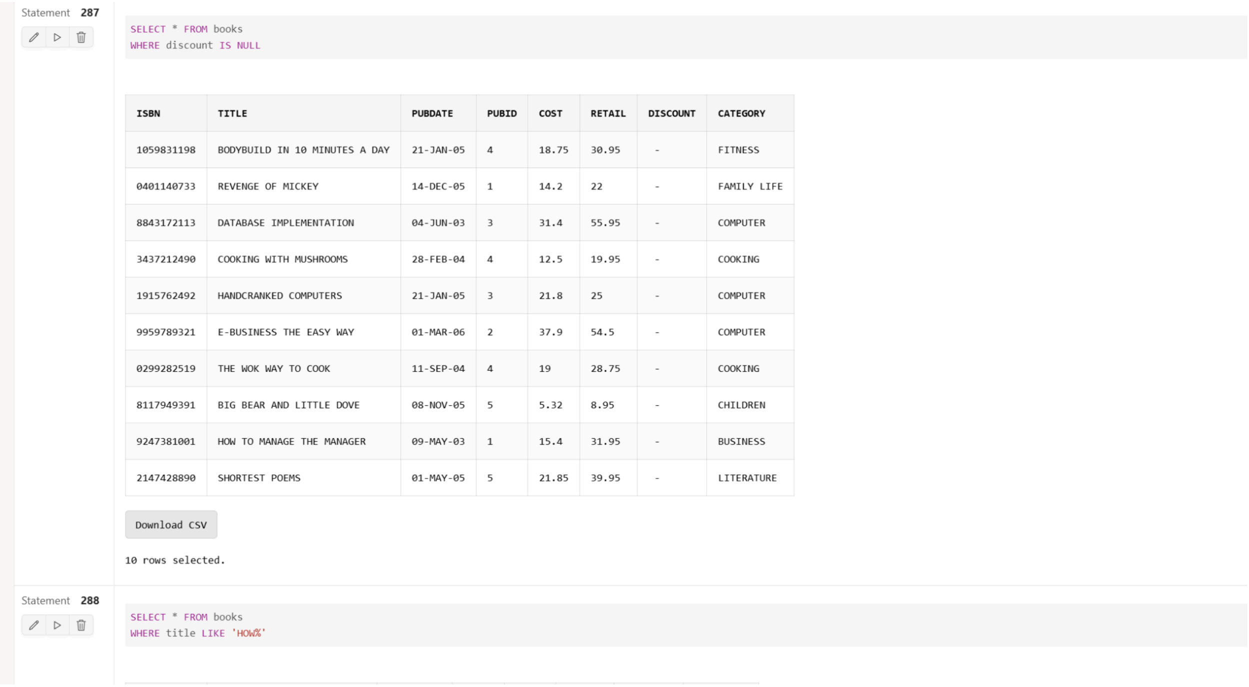
ISBN	TITLE	PUBDATE	PUBID	COST	RETAIL	DISCOUNT	CATEGORY
1059831198	BODYBUILD IN 10 MINUTES A DAY	21-JAN-05	4	18.75	30.95	-	FITNESS
8843172113	DATABASE IMPLEMENTATION	04-JUN-03	3	31.4	55.95	-	COMPUTER
3437212490	COOKING WITH MUSHROOMS	28-FEB-04	4	12.5	19.95	-	COOKING
3957136468	HOLY GRAIL OF ORACLE	31-DEC-05	3	47.25	75.95	3.8	COMPUTER
1915762492	HANDCRANKED COMPUTERS	21-JAN-05	3	21.8	25	-	COMPUTER
9959789321	E-BUSINESS THE EASY WAY	01-MAR-06	2	37.9	54.5	-	COMPUTER
0299282519	THE WOK WAY TO COOK	11-SEP-04	4	19	28.75	-	COOKING

```
7 rows selected.
```

Statement 287

```
SELECT * FROM books
WHERE discount IS NULL
```

ISBN	TITLE	PUBDATE	PUBID	COST	RETAIL	DISCOUNT	CATEGORY
1059831198	BODYBUILD IN 10 MINUTES A DAY	21-JAN-05	4	18.75	30.95	-	FITNESS
0401140733	REVENGE OF MICKEY	14-DEC-05	1	14.2	22	-	FAMILY LIFE



Statement 287

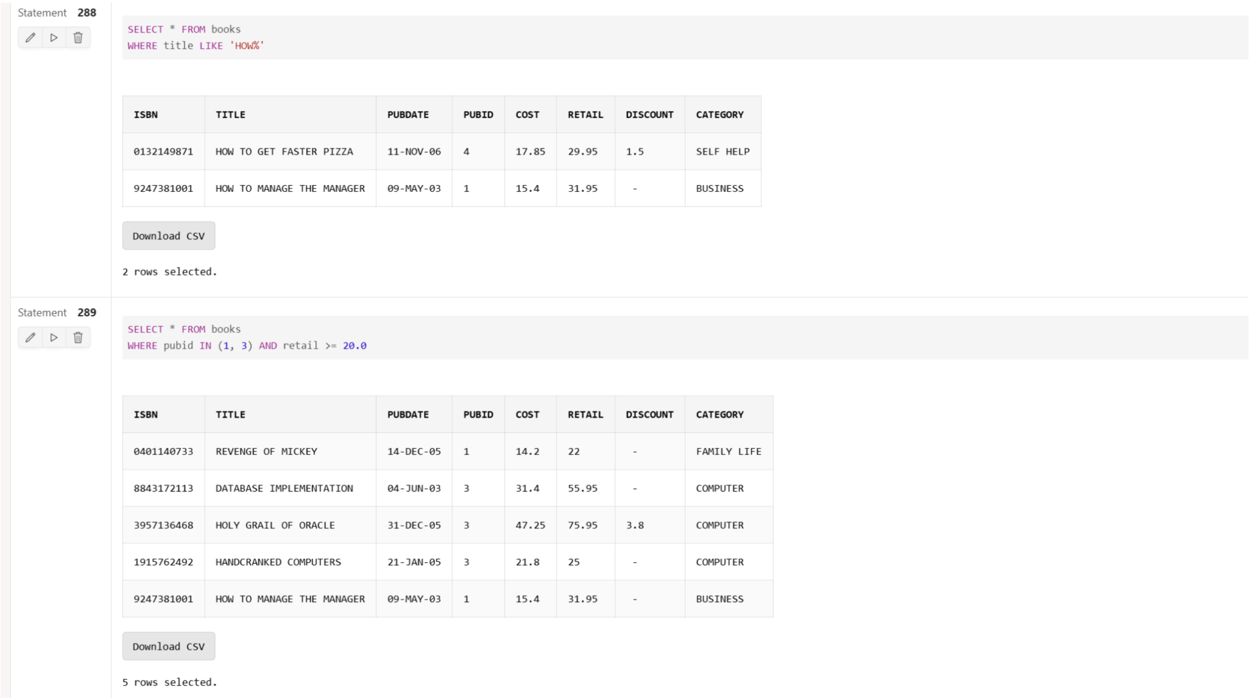
```
SELECT * FROM books
WHERE discount IS NULL
```

ISBN	TITLE	PUBDATE	PUBID	COST	RETAIL	DISCOUNT	CATEGORY
1059831198	BODYBUILD IN 10 MINUTES A DAY	21-JAN-05	4	18.75	30.95	-	FITNESS
0401140733	REVENGE OF MICKEY	14-DEC-05	1	14.2	22	-	FAMILY LIFE
8843172113	DATABASE IMPLEMENTATION	04-JUN-03	3	31.4	55.95	-	COMPUTER
3437212490	COOKING WITH MUSHROOMS	28-FEB-04	4	12.5	19.95	-	COOKING
1915762492	HANDCRANKED COMPUTERS	21-JAN-05	3	21.8	25	-	COMPUTER
9959789321	E-BUSINESS THE EASY WAY	01-MAR-06	2	37.9	54.5	-	COMPUTER
0299282519	THE WOK WAY TO COOK	11-SEP-04	4	19	28.75	-	COOKING
8117949391	BIG BEAR AND LITTLE DOVE	08-NOV-05	5	5.32	8.95	-	CHILDREN
9247381001	HOW TO MANAGE THE MANAGER	09-MAY-03	1	15.4	31.95	-	BUSINESS
2147428890	SHORTEST POEMS	01-MAY-05	5	21.85	39.95	-	LITERATURE

```
10 rows selected.
```

Statement 288

```
SELECT * FROM books
WHERE title LIKE 'HOW%'
```



Statement 288

```
SELECT * FROM books
WHERE title LIKE 'HOW%'
```

ISBN	TITLE	PUBDATE	PUBID	COST	RETAIL	DISCOUNT	CATEGORY
0132149871	HOW TO GET FASTER PIZZA	11-NOV-06	4	17.85	29.95	1.5	SELF HELP
9247381001	HOW TO MANAGE THE MANAGER	09-MAY-03	1	15.4	31.95	-	BUSINESS

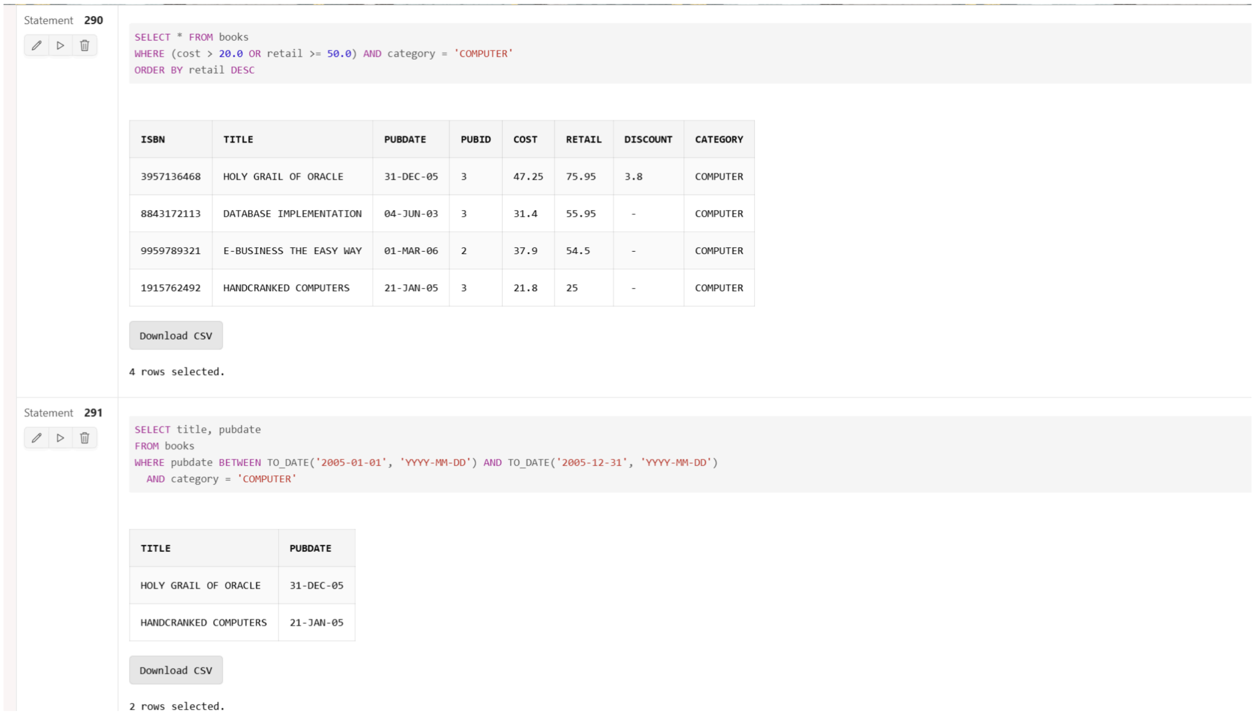
2 rows selected.

Statement 289

```
SELECT * FROM books
WHERE pubid IN (1, 3) AND retail >= 20.0
```

ISBN	TITLE	PUBDATE	PUBID	COST	RETAIL	DISCOUNT	CATEGORY
0401140733	REVENGE OF MICKEY	14-DEC-05	1	14.2	22	-	FAMILY LIFE
8843172113	DATABASE IMPLEMENTATION	04-JUN-03	3	31.4	55.95	-	COMPUTER
3957136468	HOLY GRAIL OF ORACLE	31-DEC-05	3	47.25	75.95	3.8	COMPUTER
1915762492	HANDCRANKED COMPUTERS	21-JAN-05	3	21.8	25	-	COMPUTER
9247381001	HOW TO MANAGE THE MANAGER	09-MAY-03	1	15.4	31.95	-	BUSINESS

5 rows selected.



Statement 290

```
SELECT * FROM books
WHERE (cost > 20.0 OR retail >= 50.0) AND category = 'COMPUTER'
ORDER BY retail DESC
```

ISBN	TITLE	PUBDATE	PUBID	COST	RETAIL	DISCOUNT	CATEGORY
3957136468	HOLY GRAIL OF ORACLE	31-DEC-05	3	47.25	75.95	3.8	COMPUTER
8843172113	DATABASE IMPLEMENTATION	04-JUN-03	3	31.4	55.95	-	COMPUTER
9959789321	E-BUSINESS THE EASY WAY	01-MAR-06	2	37.9	54.5	-	COMPUTER
1915762492	HANDCRANKED COMPUTERS	21-JAN-05	3	21.8	25	-	COMPUTER

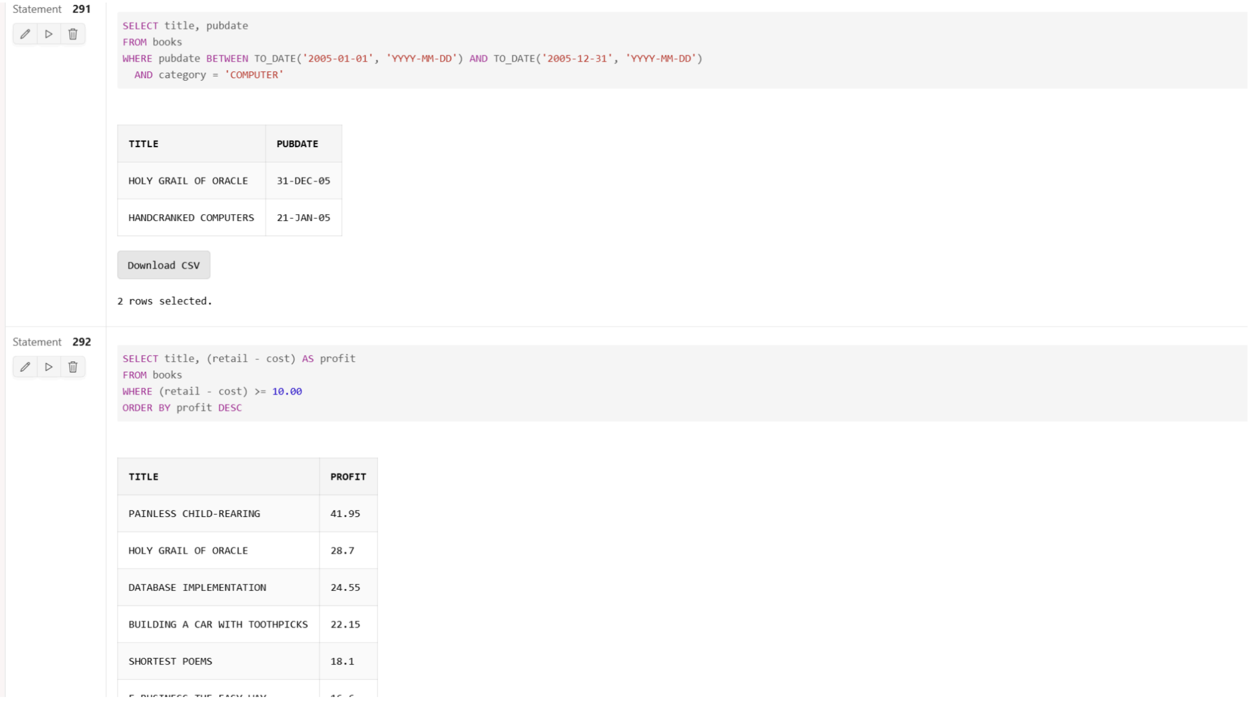
4 rows selected.

Statement 291

```
SELECT title, pubdate
FROM books
WHERE pubdate BETWEEN TO_DATE('2005-01-01', 'YYYY-MM-DD') AND TO_DATE('2005-12-31', 'YYYY-MM-DD')
  AND category = 'COMPUTER'
```

TITLE	PUBDATE
HOLY GRAIL OF ORACLE	31-DEC-05
HANDCRANKED COMPUTERS	21-JAN-05

2 rows selected.



Statement 291

```
SELECT title, pubdate
FROM books
WHERE pubdate BETWEEN TO_DATE('2005-01-01', 'YYYY-MM-DD') AND TO_DATE('2005-12-31', 'YYYY-MM-DD')
  AND category = 'COMPUTER'
```

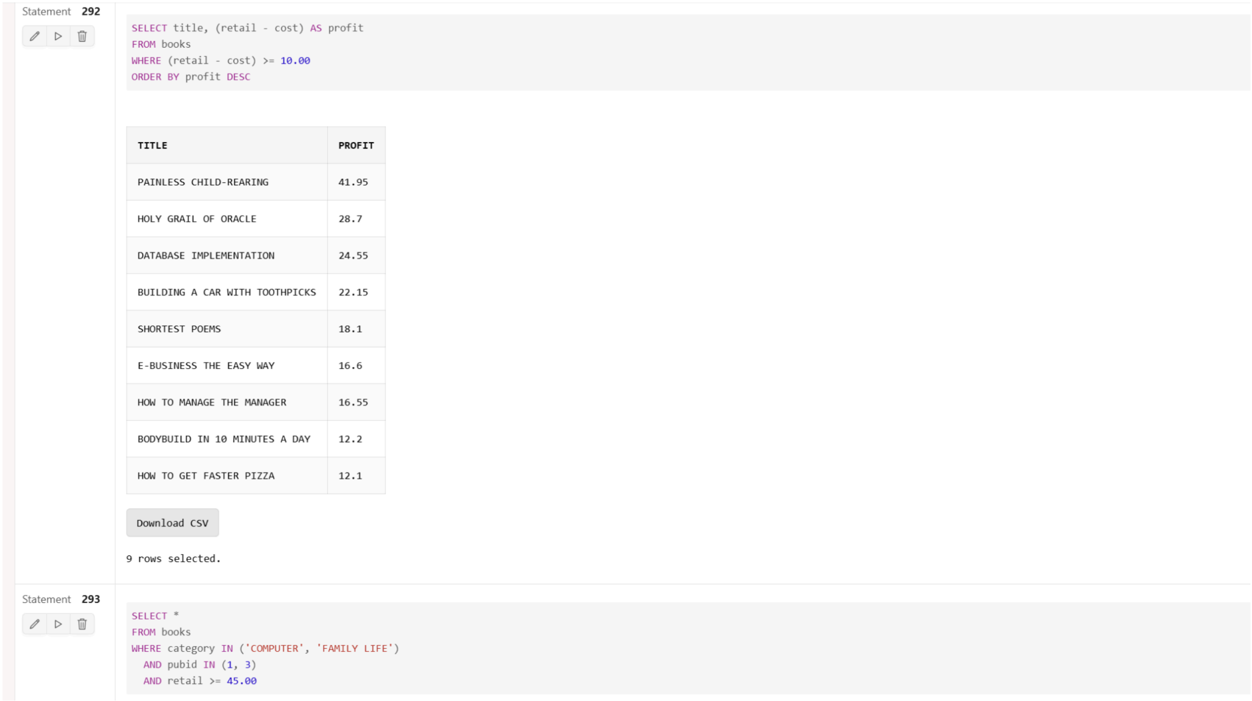
TITLE	PUBDATE
HOLY GRAIL OF ORACLE	31-DEC-05
HANDCRANKED COMPUTERS	21-JAN-05

2 rows selected.

Statement 292

```
SELECT title, (retail - cost) AS profit
FROM books
WHERE (retail - cost) >= 10.00
ORDER BY profit DESC
```

TITLE	PROFIT
PAINLESS CHILD-REARING	41.95
HOLY GRAIL OF ORACLE	28.7
DATABASE IMPLEMENTATION	24.55
BUILDING A CAR WITH TOOTHPICKS	22.15
SHORTEST POEMS	18.1
E-BUSINESS THE EASY WAY	16.6

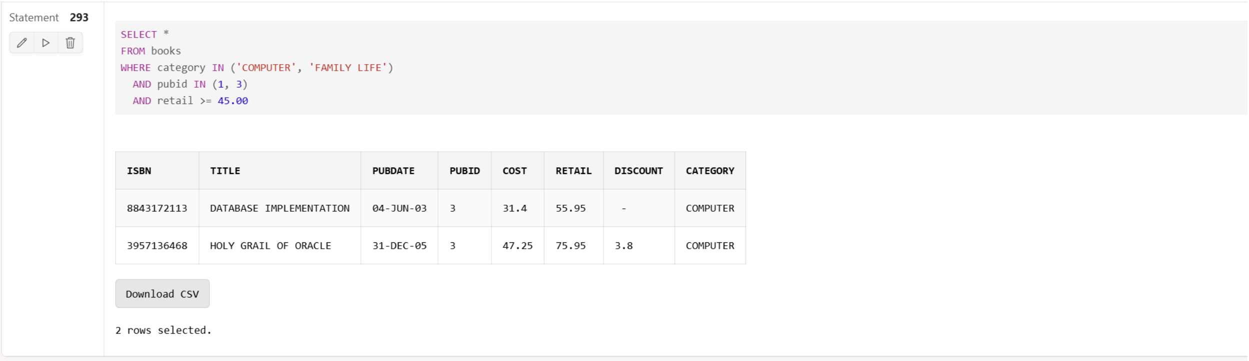


```
SELECT title, (retail - cost) AS profit
FROM books
WHERE (retail - cost) >= 10.00
ORDER BY profit DESC
```

TITLE	PROFIT
PAINLESS CHILD-REARING	41.95
HOLY GRAIL OF ORACLE	28.7
DATABASE IMPLEMENTATION	24.55
BUILDING A CAR WITH TOOTHPICKS	22.15
SHORTEST POEMS	18.1
E-BUSINESS THE EASY WAY	16.6
HOW TO MANAGE THE MANAGER	16.55
BODYBUILD IN 10 MINUTES A DAY	12.2
HOW TO GET FASTER PIZZA	12.1

9 rows selected.

```
SELECT *
FROM books
WHERE category IN ('COMPUTER', 'FAMILY LIFE')
  AND pubid IN (1, 3)
  AND retail >= 45.00
```



```
SELECT *
FROM books
WHERE category IN ('COMPUTER', 'FAMILY LIFE')
  AND pubid IN (1, 3)
  AND retail >= 45.00
```

ISBN	TITLE	PUBDATE	PUBID	COST	RETAIL	DISCOUNT	CATEGORY
8843172113	DATABASE IMPLEMENTATION	04-JUN-03	3	31.4	55.95	-	COMPUTER
3957136468	HOLY GRAIL OF ORACLE	31-DEC-05	3	47.25	75.95	3.8	COMPUTER

2 rows selected.

1. SQL Query with Arithmetic Comparison Operators

The first task was to retrieve records from the `JustLee Books` database using one or more arithmetic comparison operators (e.g., `=`, `!=`, `>`, `<`, `>=`). In the example query, the `cost` column was filtered to show only books that had a value greater than 20.0. This allows us to find books with a cost exceeding a specific threshold, which is useful for understanding the pricing distribution or selecting premium books.

---

2. SQL Query with `BETWEEN ... AND` Operator

This query uses the `BETWEEN ... AND` operator to find records within a specific range. The example focuses on retrieving books published between two specific dates. By using `BETWEEN`, the query ensures that only books published between the start and end date are included, which can be useful for analyzing books released during a certain time frame, such as a year or a specific publication period.

---

3. SQL Query with the `IN` Operator

In this task, the `IN` operator is used to filter rows based on multiple possible values for the `category` column. By specifying categories like `'FITNESS'`, `'COOKING'`, and `'COMPUTER'`, we can retrieve records that belong to any of these groups. This type of filtering is beneficial for finding books that are part of different but relevant genres, helping to group similar products.

---

4. SQL Query with the `LIKE` Operator

The query here uses the `LIKE` operator with the `%` wildcard to find records where the `title` starts with the word `'HOW'`. The `%` allows for any sequence of characters after `'HOW'`. This query is particularly useful for pattern matching, such as searching for books with similar titles or identifying books that belong to a specific series.

---

5. SQL Query with `IS NULL` Operator

The `IS NULL` operator was used to retrieve records where a specific column (`discount`) has no value assigned (`NULL`). This helps in identifying entries that lack certain information, which can be important for data cleaning or highlighting books that might need updated or additional information.

---

6. SQL Query with Multiple Conditions (`AND` and `OR` Logical Operators)

In this example, multiple conditions are combined using `AND` and `OR` logical operators. The query filters records to find books published by specific publishers (`pubid` of 1 or 3) and have a retail price of at least $20.00. Combining conditions allows for more precise data retrieval, which is useful for generating complex reports or targeting specific segments of data.

---

7. Complex SQL Query with Logical Operators and `ORDER BY`

This query involves a complex set of conditions, using both arithmetic and special operators, and includes an `ORDER BY` clause to sort the results. The `ORDER BY` ensures that the data is presented in descending order based on retail price. Such queries are valuable for analyzing data based on specific attributes and then organizing it for better visibility, such as displaying the most expensive books first.

---

8. Retrieve Titles and Publish Dates of Books Published in 2005 (Computer Category)

The query uses a range condition (`BETWEEN`) to filter books published in 2005, specifically those in the `'COMPUTER'` category. By restricting results to this year, the query helps in focusing on books released in a particular time frame, aiding in trend analysis or understanding publication patterns for computer-related topics.

---

9. Retrieve Titles of Books Generating a Profit of At Least $10.00

In this query, the calculated profit (`retail - cost`) is used to filter books generating a profit of at least $10.00. The results are sorted in descending order by profit to identify the most profitable books. This kind of analysis is crucial for understanding which products contribute most to revenue and can help in decision-making for pricing strategies.

---

10. Retrieve Books in Specific Categories and Published by Specific Publishers, Price Greater Than $45.00

The final query retrieves books from either the `'COMPUTER'` or `'FAMILY LIFE'` categories that are published by `Publisher 1` or `Publisher 3`. Additionally, only books priced at $45.00 or above are included. This helps filter out less expensive books and focus on those that meet both category and publisher criteria, which can be helpful for targeted marketing or stock management.