Anurag Bodkhe

ADT23SOCB1509

Assignment-6

**Study and Implementation of SQL Clauses**

SQL **clauses** help in filtering, grouping, and sorting data for efficient data retrieval. Below, we will implement the **GROUP BY & HAVING clause, ORDER BY clause, and Indexing** with examples.

**1) GROUP BY & HAVING Clause**

* GROUP BY **groups rows with the same values** in specified columns.
* HAVING **filters grouped records** (similar to WHERE, but for grouped results).

**Example: Count books by Author using GROUP BY**

SELECT Author, COUNT(\*) AS Total\_Books

FROM Book

GROUP BY Author;

📌 **Explanation:**

* Groups books by **Author** and counts the number of books per author.

**Example: Use HAVING to filter groups (Books with more than 1 book per author)**

SELECT Author, COUNT(\*) AS Total\_Books

FROM Book

GROUP BY Author

HAVING COUNT(\*) > 1;

📌 **Explanation:**

* Only displays authors who have **more than one book** in the database.

**2) ORDER BY Clause**

* ORDER BY **sorts query results** in ascending (ASC) or descending (DESC) order.

**Example: Sorting books by Year\_Published (Oldest to Newest)**

SELECT Title, Author, Year\_Published

FROM Book

ORDER BY Year\_Published ASC;

📌 **Explanation:**

* Retrieves all books sorted by **Year Published (Ascending Order: oldest first)**.

**Example: Sorting members by Name in Descending Order**

SELECT Member\_ID, Name, Membership\_Date

FROM Member

ORDER BY Name DESC;

📌 **Explanation:**

* Retrieves all members sorted by **Name in Descending Order (Z → A)**.

**3) Indexing in SQL**

* **Indexes** improve **query performance** by speeding up searches.
* An **index** is automatically created on **Primary Key** columns, but you can manually index other frequently used columns.

**Example: Creating an Index on Author column in Book table**

CREATE INDEX idx\_Author ON Book (Author);

📌 **Explanation:**

* This creates an index **idx\_Author** on the Author column, making searches **faster**.

**Example: Creating a Unique Index on Email column in Member table**

CREATE UNIQUE INDEX idx\_Email ON Member (Email);

📌 **Explanation:**

* Ensures that **no duplicate emails** exist while improving search performance.

**Example: Dropping an Index**

DROP INDEX idx\_Author ON Book;

📌 **Explanation:**

* Removes the **idx\_Author** index from the **Book** table.

**Summary of SQL Clauses:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clause** | |  | | --- | |  |  |  | | --- | | **Example** | | **Purpose** |
| GROUP BY | |  | | --- | |  |  |  | | --- | | GROUP BY Author | | |  | | --- | |  |  |  | | --- | | Groups records by Author | |
| HAVING | HAVING COUNT(\*) > 1 | |  | | --- | |  |  |  | | --- | | Filters grouped results | |
| ORDER BY | |  | | --- | |  |  |  | | --- | | ORDER BY Year\_Published DESC | | Sorts results Descending |
| INDEX | CREATE INDEX idx\_Author ON Book(Author) | Improves search performance |