DBI 2024-2025: Quiz Feedback

COMP1048: Databases and Interfaces (2024-2025)

Matthew Pike & Yuan Yao

General Feedback

Overall, the SQL quiz submissions demonstrated a good understanding of SQL concepts and syntax. However, there were some common errors and misconceptions that appeared across multiple questions. These included syntax errors, incorrect use of SQL clauses/functions, logical errors, and ambiguities/missing specifications. Below is a summary of the common issues identified in the quiz submissions:

Syntax Errors (The most common error among all questions)

- Missing comma (`,`) between column names in the `SELECT` statement.
- Forgetting to include `;` at the end of the SQL query.
- Misusing punctuation in the wrong places (e.g., using a semicolon in the middle of a SQL query).
- Spelling errors in keywords, column names, or table names (e.g., misspelt `TrackId` as `Trackld`, spelt `JOIN` as `JION`, spelt `ORDER BY` as `ORDERED BY`).
  - Note: A penalty of 0.5 marks per character is applied for each misspelt column name. We explicitly mentioned that the column names should be spelled correctly and provided clear requirements for each question.

Incorrect Use of SQL Clauses/Functions

- Confused the usage of `WHERE` and `HAVING`.
  - `WHERE` is used to filter rows before grouping.
  - `HAVING` is used to filter groups after grouping.
- Misuse of `LIMIT` when it should be a `WHERE` condition (e.g., `LIMIT 100` instead of `WHERE Milliseconds < 100`).
- Incorrect placement of `ORDER BY`, `GROUP BY`, and `WHERE`; lacked a clear understanding of their order.
- Confused using `JOIN` and `WHERE ... IN`.
- Used the wrong JOIN type, especially confusing `INNER JOIN`, `NATURAL JOIN`, and `LEFT JOIN`.
- Did not specify the table when selecting columns from multiple tables.
- When using aggregate functions such as `SUM()` or `COUNT()`, misused `WHERE` instead of `HAVING` when doing the filtering.

Logical Errors

- Tried to select a column that doesn't exist in the specified table.
- Used incorrect conditions for filtering or grouping data, such as selecting by names instead of IDs.

Ambiguities and Missing Specifications

- Did not specify the table name when selecting columns from multiple tables.
- Left statements incomplete (e.g., missing `FROM` when using `SELECT`).

Question-Specific Feedback

Q1

Q1 was a check to ensure you were using the correct version of SQLite. There were no marks associated with this question.

Q2

Misspelling was the most common issue, such as using `ld` instead of `Id` for column names. Other errors include incorrectly using a table name when renaming and syntax issues like writing `SELECT GenreId and Name`.

Q3

Many solutions misused comparison operators, such as writing `>100,0000` instead of the correct condition `≤ 1000000`. Another common mistake was using `*` to select all columns rather than specifying the required ones. Additionally, some solutions demonstrated a misunderstanding of the term/concept of a track.

Q4

A common error was the incorrect use of the logical operator `OR`, such as writing `WHERE ArtistId=1 OR 2` instead of `WHERE ArtistId=1 OR ArtistId=2`. Many solutions also failed to order results, despite the question explicitly requested an ordered output.

Q5

The common error was using `WHERE` instead of `HAVING` when applying it to an aggregate function. Many solutions incorrectly used `WHERE` instead of `GROUP BY`. Additionally, some solutions misused aggregate functions like `COUNT()`, or incorrectly wrote conditions such as `COUNT = 10` instead of `COUNT ≤ 10` or `COUNT ≥ 10`.

Q6

The most common error was using `WHERE ... IN` instead of `JOIN`. Many solutions also confused table references, selecting columns without specifying the table they belong to. Additionally, many solutions incorrectly joined tables on columns that do not represent valid relationships. Some solutions used `NATURAL JOIN` or `LEFT JOIN` instead of `INNER JOIN`, or omitted `INNER JOIN` altogether. Others used an incorrect select condition, resulting in an empty output.

Q7

A common error was not knowing when to use `COUNT()` versus `SUM()`. Some solutions failed to use `LIMIT` to restrict the number of output rows, while others omitted `GROUP BY`. Additionally, there were minor typos when renaming columns.

Q8

Many solutions misused `LEFT JOIN` instead of the required `INNER JOIN`. There was also frequent misuse of `COUNT()` and `SUM()`, often applying them to incorrect columns or using them in the wrong context. Additionally, some solutions misused `WHERE` instead of `HAVING`.

Q9

A common error was the misuse of `COUNT()` and `SUM()` or failing to use `JOIN` correctly, including the use of incorrect `JOIN` types. Many solutions also confused `WHERE` and `HAVING`. Several misplaced clauses like `GROUP BY`, `LIMIT`, or `ORDER BY`, resulting in incorrect outputs. Another frequent issue was the omission of required table specifications, leading to ambiguous queries. Additionally, there was a tendency to overcomplicate queries with unnecessary constraints or functions.

Q10

The most common error was confusion over how to convert milliseconds to minutes. Another issue was incorrectly applying functions like `SUM()` and `ROUND()` in the wrong order or in the wrong context. Some solutions used `GROUP BY` on `name` instead of `PlaylistId`, while others failed to round values to 0 decimal places.

Q11

The most common error was a misunderstanding of the question requirements. Some solutions also misused `WHERE` when it was not needed.