SAMPLE SQL Questions

Sample Answers

Sample Questions for Century Inventions Database

Century_Inventions_SampleTasks_wSampleAnswers.docx

Unit Code/s & Name/s	ICTPRG431 Apply query language in relational databases
Activity Name	SQL – Sample Questions

In preparation for ICTPRG431 (Apply Queries) - AT2

The following questions reference the **century_inventions** database.

Q1: Write an SQL statement to retrieve all inventions

- Sort the results by year in descending order.
- Provide a screen image or copy of the query, and the first 4 records of the results.

```
SELECT TOP (100) PERCENT invention_description, year
FROM dbo.invention
ORDER BY year DESC
```

```
self cleaning kitchens                                          2021
Self Cleaning Surfaces                                          2019
building miniature machines out of molecules                    2015
announces "hyperloop" a giant, pneumatic tube transport system 2013
```

Q2: Write an SQL statement to retrieve all inventions using the following filters:

- Year before 1920
- Invention_description contains the word “self”. (Use the LIKE operator).

Sort the results by year in descending order. Provide a screen image of the query and the results.

```
SELECT      TOP (100) PERCENT invention_description, year
FROM         dbo.invention
WHERE       (invention_description LIKE N'%Self%')
ORDER BY year DESC
```

```
self cleaning kitchens    2021
Self Cleaning Surfaces 2019
```

Q3: Use an aggregate function to write an SQL statement to retrieve the number of inventions grouped by Year.
Provide a screen image of the query and the first 10 or so results (approx.).

```
SELECT       COUNT(*) AS Expr1, year
FROM         dbo.invention
GROUP BY year
ORDER BY year
```

```
1 1006
1 1009
1 1019
1 1020
1 1901
1 1903
2 1905
2 1906
2 1907
1 1908
2 1912
1 1916
4 1920
1 1921
```

Q4: Write an SQL statement to retrieve all inventors and their associated inventions, using the following filters:

- Inventor nationality is either Italian or Belgian.
- The invention was made between 1900 and 1910

Sort the results by inventor name. Provide a screen image of the query and the results.

```
SELECT       TOP (100) PERCENT dbo.inventor.inventor_name, dbo.invention.invention_description,
dbo.inventor.nationality, dbo.invention.year
FROM         dbo.invention INNER JOIN
               dbo.invention_list ON dbo.invention.invention_id = dbo.invention_list.invention_id INNER JOIN
               dbo.inventor ON dbo.invention_list.inventor_id = dbo.inventor.inventor_id
WHERE       (dbo.inventor.nationality = N'Italian') AND (dbo.invention.year BETWEEN 1900 AND 1910) OR
               (dbo.inventor.nationality = N'Belgian') AND (dbo.invention.year BETWEEN 1900 AND 1910)
ORDER BY dbo.inventor.inventor_name
```

```
Guglielmo Marconi    radio waves across Atlantic Ocean    Italian    1901
Leo Baekeland        Bakelite (synthetic plastic          Belgian    1907
```

Q5: Write an SQL statement to retrieve all inventors

- Sort the results by nationality and then by inventor name in ascending order.
- Provide a screen image or copy of the query, and the last 4 records of the results.

```
SELECT       TOP (100) PERCENT dbo.inventor.nationality, dbo.inventor.inventor_name
FROM          dbo.invention INNER JOIN
                  dbo.invention_list ON dbo.invention.invention_id = dbo.invention_list.invention_id INNER
JOIN
                  dbo.inventor ON dbo.invention_list.inventor_id = dbo.inventor.inventor_id
ORDER BY dbo.inventor.nationality, dbo.inventor.inventor_name
```

Hungarian-Argentinian	Laszlo Biro
Italian	Guglielmo Marconi
Russian-Italian	Mikhail Tswett
Scottish	John Logie Baird

Q6: Make a duplicate of your Q5. Change the new query, so that the query now:

- Groups the inventors by nationality and
- Counts the number of inventors for each nationality.

Provide a screen image or copy of the query, and the last 4 records of the results.

```
SELECT       TOP (100) PERCENT dbo.inventor.nationality, COUNT(dbo.inventor.inventor_name) AS
Expr1
FROM          dbo.invention INNER JOIN
                  dbo.invention_list ON dbo.invention.invention_id = dbo.invention_list.invention_id INNER
JOIN
                  dbo.inventor ON dbo.invention_list.inventor_id = dbo.inventor.inventor_id
GROUP BY dbo.inventor.nationality
ORDER BY dbo.inventor.nationality
```

Hungarian-Argentinian	1
Italian	1
Russian-Italian	1
Scottish	1

Q7: Make a further duplicate of your Q5. Change the new query, so that the query now:

- Concatenate the inventors so the query only displays a row per inventors for each nationality and
- Use the following alias for the list of inventors: Inventors.

Provide a screen image or copy of the query, and the first 4 records of the results.

(Reworded question: Use an aggregate function to write an SQL statement to retrieve the list of inventors by nationality.)

```
SELECT …,
GROUP_CONCAT(inventor.inventor_name) AS Inventors …
```

The SQL Server Equivalent to GROUP_CONCAT()

https://database.guide/the-sql-server-equivalent-to-group_concat/

Before SQL Server 2017 came along, there wasn't a T-SQL equivalent of the MySQL `GROUP_CONCAT()` function. This function allows you to return a result set as a comma-separated list, as opposed to listing each row as a separate row (as with a normal result set).

Prior to SQL Server 2017, if you wanted to put your result into a comma separated list, you'd need to find a workaround, perhaps using a combination of `STUFF()`, `FOR XML`, and `PATH()`.

However, T-SQL now has the `STRING_AGG()` function which is available from SQL Server 2017. This function does pretty much the same thing as MySQL's `GROUP_CONCAT()` function (with some minor differences).

Syntax

The syntax of the `STRING_AGG()` function goes like this:

```
STRING_AGG ( expression, separator ) [ <order_clause> ]

<order_clause> ::=
    WITHIN GROUP ( ORDER BY <order_by_expression_list> [ ASC | DESC ] )
```

Where `expression` is an expression of any type. Expressions are converted to `NVARCHAR` or `VARCHAR` types during concatenation. Non-string types are converted to `NVARCHAR` type.

Where separator is an expression of `NVARCHAR` or `VARCHAR` type that is used as separator for concatenated strings. It can be literal or variable.

The (optional) order clause consists of `WITHIN GROUP` followed by `ORDER BY ASC` or `ORDER BY DESC` in parentheses. `ASC` orders the result in ascending order. This is the default value. `DESC` orders the result in descending order.

Example

Here's a quick example of the `STRING_AGG()` function:

```
SELECT STRING_AGG(Genre, ',') AS Result
FROM Genres;
```

Result:

```
Result
--------------------------------------------
Rock,Jazz,Country,Pop,Blues,Hip Hop,Rap,Punk
```

As you can see, the result set is listed out as a comma separated list. This is because our second argument is a comma, which specifies that a comma should be used as the separator.

Q8. Write an SQL statement that displays the list of inventions, where the invention's record has been modified since the original record was added.

- IE: the Date Created is different to the Date Modified in the Changelog table.
- Sort by invention description.

Provide a screen image or copy of the query, and the first 4 records of the results.

```
SELECT      TOP (100) PERCENT dbo.invention.invention_description, dbo.changelog.date_created,
dbo.changelog.date_modified
FROM            dbo.invention INNER JOIN
                  dbo.changelog ON dbo.invention.invention_id = dbo.changelog.invention_id
WHERE       (dbo.changelog.date_created <> dbo.changelog.date_modified)
ORDER BY dbo.invention.invention_description
```

```
agreement to make Wi-Fi a worldwide standard for wireless Internet   2021-05-19 2021-06-16
MEGA 1, believed to be the world's first radio-controlled wristwatch. 2021-04-14 2021-04-18
Space Shuttle makes its maiden voyage                                 2021-02-05 2021-02-07
announces "hyperloop" a giant, pneumatic tube transport system        2021-07-02 2021-07-11
```

Q9. Use an IIF control statement to present a list of Invention_IDs, followed by the words: ‘Equal’ or ‘Not Equal’ dpending on whether the InventionID is the same as (equal to) the Inventor_ID within the Invention_List table.

Provide a screen image or copy of the query, and the first 20 records of the results.

```
SELECT      invention_id, IIF(inventor_id = invention_id, 'Equal','Not Equal') AS MyIDComparison
FROM        dbo.invention_list
ORDER BY    invention_id
```