

Nama: Aliriza Hamonangan Matondang

SQL Query SYNTAX

Instruction:

- Please use the DBeaver application and the DBeaver Sample Database (SQLite) to do the task on the questions in PART I.
- Please attach the screenshots to answer the questions. Example:

Question : Show all songs (tracks) contained in the database.

Answer:

```
SELECT * FROM track;
```

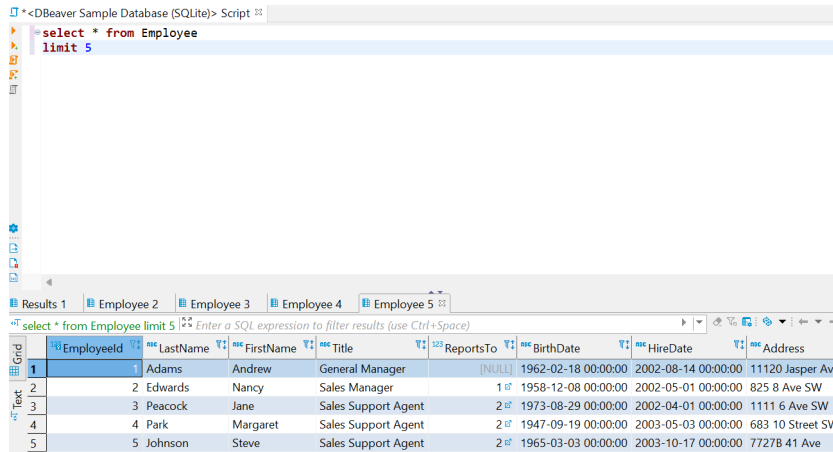
Here are the details of the tables that will be used:



Questions:

1. Show all columns of the first five rows in the employee table (employees)

Answer:

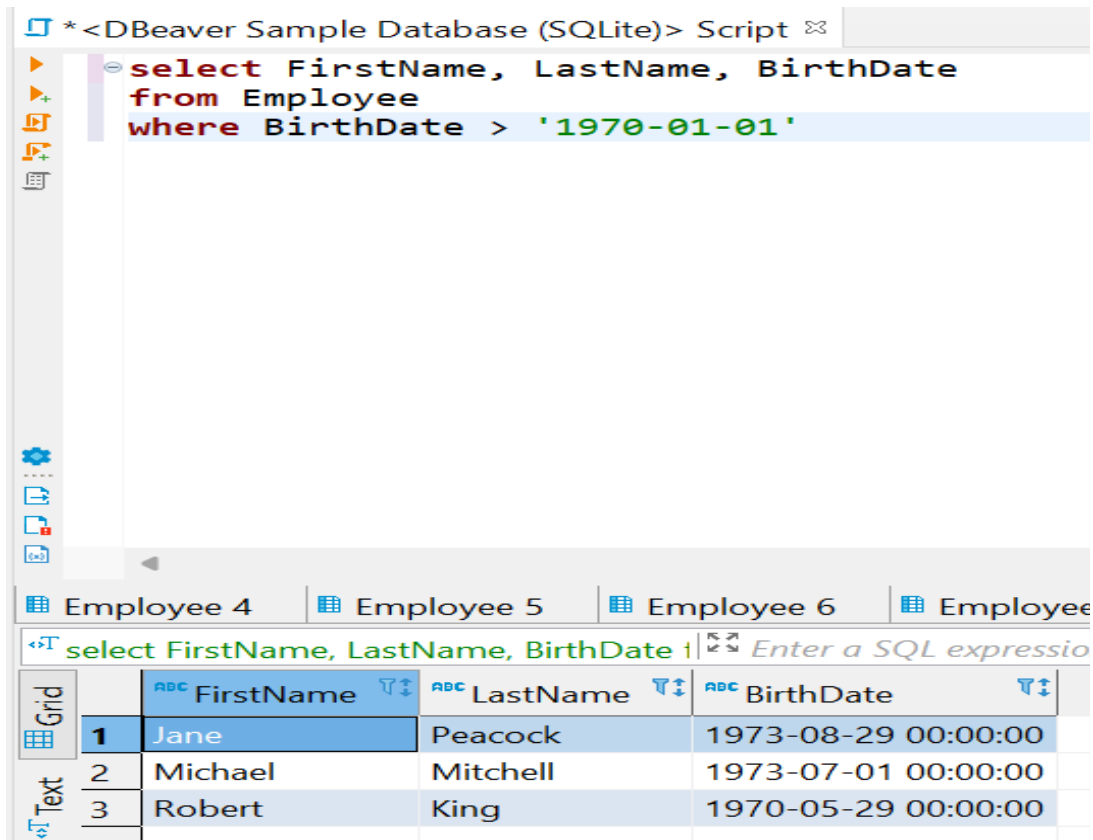


The screenshot shows the DBeaver SQL editor with the query `select * from Employee limit 5` executed. The results are displayed in a table with 8 columns: EmployeeId, LastName, FirstName, Title, ReportsTo, BirthDate, HireDate, and Address. The first five rows of data are shown.

EmployeeId	LastName	FirstName	Title	ReportsTo	BirthDate	HireDate	Address
1	Adams	Andrew	General Manager	(NULL)	1962-02-18 00:00:00	2002-08-14 00:00:00	11120 Jasper Ave
2	Edwards	Nancy	Sales Manager	1	1958-12-08 00:00:00	2002-05-01 00:00:00	825 8 Ave SW
3	Peacock	Jane	Sales Support Agent	2	1973-08-29 00:00:00	2002-04-01 00:00:00	1111 6 Ave SW
4	Park	Margaret	Sales Support Agent	2	1947-09-19 00:00:00	2003-05-03 00:00:00	683 10 Street SW
5	Johnson	Steve	Sales Support Agent	2	1965-03-03 00:00:00	2003-10-17 00:00:00	7727B 41 Ave

2. Show the first name, last name, and date of birth of employees (employees) who were born after 1970-01-01.

Answer:



The screenshot shows the DBeaver SQL editor with the query `select FirstName, LastName, BirthDate from Employee where BirthDate > '1970-01-01'` executed. The results are displayed in a table with 3 columns: FirstName, LastName, and BirthDate. The first three rows of data are shown.

FirstName	LastName	BirthDate
Jane	Peacock	1973-08-29 00:00:00
Michael	Mitchell	1973-07-01 00:00:00
Robert	King	1970-05-29 00:00:00

3. Show first name, company, and email of customers (customers) who use gmail. (Hint: email ends with '@gmail.com').

Answer:

* <DBeaver Sample Database (SQLite)> Script

```
select FirstName, Company, Email
from customer
where Email Like '%@gmail.com%'
```

Employee 2 Employee 3 Employee 4 Employee 5 Err

select FirstName, Company, Email from Enter a SQL expression to filter re

	Grid	ABC FirstName	ABC Company	ABC Email
1		François	[NULL]	ftremblay@gmail.com
2		Helena	[NULL]	hholy@gmail.com
3		Heather	[NULL]	hleacock@gmail.com
4		Frank	[NULL]	fralston@gmail.com
5		Julia	[NULL]	jubarnett@gmail.com
6		Martha	[NULL]	marthasilk@gmail.com
7		Dominique	[NULL]	dominiquelefebvre@gmail.com
8		Phil	[NULL]	phil.hughes@gmail.com

4. Show invoice ID, invoice date, and total invoices created between January 1, 2007 and December 12, 2007 and have a total of more than 13. Show the total in order of the largest to the smallest.

Answer:

*<DBeaver Sample Database (SQLite)> Script

```
select InvoiceId, InvoiceDate, Total
from Invoice
where (InvoiceDate BETWEEN '2007-01-01' AND '2007-12-12')
AND Total > 13
order by total DESC
```

Employee 7 | Employee 8 | Employee 9 | Invoice 10 | Invoice 11

select InvoiceId, InvoiceDate, Total from

	InvoiceId	InvoiceDate	Total
1	5	2007-01-11 00:00:00	13.86
2	12	2007-02-11 00:00:00	13.86
3	19	2007-03-14 00:00:00	13.86
4	26	2007-04-14 00:00:00	13.86
5	33	2007-05-15 00:00:00	13.86
6	40	2007-06-15 00:00:00	13.86
7	47	2007-07-16 00:00:00	13.86
8	54	2007-08-16 00:00:00	13.86
9	61	2007-09-16 00:00:00	13.86
10	68	2007-10-17 00:00:00	13.86

5. Show ID, first name, and last name of employees who handle at least one customer (SupportRepId) (Hint: Use EXISTS).

Jawaban:

*<DB Beaver Sample Database (SQLite)> Script

```
select EmployeeId, FirstName, LastName
FROM Employee e
where EXISTS (SELECT SupportRepId
              from Customer
              where CustomerId = e.EmployeeId and SupportRepId >= 1)
```

Employee 22 Employee 23 Employee 24 Results 25 Results 26 Emplo

select EmployeeId, FirstName, LastName Enter a SQL expression to filter results (use Ctrl+Space)

	EmployeeId	FirstName	LastName
1	1	Andrew	Adams
2	2	Nancy	Edwards
3	3	Jane	Peacock
4	4	Margaret	Park
5	5	Steve	Johnson
6	6	Michael	Mitchell
7	7	Robert	King
8	8	Laura	Callahan

6. Show full name and employee category with the following conditions.
- Full name: Combine of first name and last name separated by a space, example: First Name = 'Andrew', Last Name = 'Adams' → Full Name = 'Andrew Adams'
 - The employee category consists of two categories, namely 'staff' and 'manager'. All employees who have titles containing the word 'manager' will enter the 'manager' category, while the others will enter the 'staff' category.

Answer:

The screenshot shows the DBeaver SQL editor with the following query:

```
select (FirstName || ' ' || LastName) as FullName,
case
    when Title like '%Manager%' then 'Manager'
    when Title like '%staff%' then 'Staf'
end 'Employee_Category'
from Employee
```

Below the editor, the results grid is displayed with the following data:

	FullName	Employee_Category
1	Andrew Adams	Manager
2	Nancy Edwards	Manager
3	Jane Peacock	[NULL]
4	Margaret Park	[NULL]
5	Steve Johnson	[NULL]
6	Michael Mitchell	Manager
7	Robert King	Staf
8	Laura Callahan	Staf

7. Show Invoice ID, total unit price, and total quantity for each invoice from the InvoiceLine table without JOIN with other tables.

Answer:

*<DBeaver Sample Database (SQLite)> Script

```
select InvoiceId, sum(UnitPrice) as total_unit_price, sum(Quantity) as total_quantity
from InvoiceLine
group by InvoiceId
limit 10
```

Employee 28 | Employee 29 | Results 30 | Results 31 | Results 32 | Results 33 | Results 34 | Inv

select InvoiceId, sum(UnitPrice) as tot: Enter a SQL expression to filter results (use Ctrl+Space)

	InvoiceId	total_unit_price	total_quantity
1	1	1.98	2
2	2	3.96	4
3	3	5.94	6
4	4	8.91	9
5	5	13.86	14
6	6	0.99	1
7	7	1.98	2
8	8	1.98	2
9	9	3.96	4
10	10	5.94	6

8. Show the Invoice ID and the total price that has an even ID, sorted from the smallest to the largest ID.

Answer:

```
SELECT InvoiceId, sum(UnitPrice) as total_unit_price
from InvoiceLine
where InvoiceId %2 = 0
group by InvoiceId
order by InvoiceId
limit 10
```

Results 1 | InvoiceLine 2 | InvoiceLine 3 | InvoiceLine 4

SELECT InvoiceId, sum(UnitPrice) as *Enter a SQL expression to filter results*

Grid	123 InvoiceId	123 total_unit_price
1	2	3.96
2	4	8.91
3	6	0.99
4	8	1.98
5	10	5.94
6	12	13.86
7	14	1.98
8	16	3.96
9	18	8.91
10	20	0.99

9. Show all invoices ID that have an average unit price of more than 0.5 (Hint: using HAVING).

Answer:

* <DBeaver Sample Database (SQLite)> Script

```
select InvoiceId, round(avg(UnitPrice),2) avg_UnitPrice
from InvoiceLine
group by InvoiceId
HAVING avg_UnitPrice > 0.5
```

Results 30 Results 31 Results 32 Results 33 Results 34

select InvoiceId, round(avg(UnitPrice),2) avg_UnitPrice

	InvoiceId	avg_UnitPrice
1	1	0.99
2	2	0.99
3	3	0.99
4	4	0.99
5	5	0.99
6	6	0.99
7	7	0.99
8	8	0.99
9	9	0.99
10	10	0.99

10. Show employee ID, employee first name, and customer first name that have been handled by the employee. Change the name of the column that is displayed if there are two columns with the same name.

Answer:

*<DBeaver Sample Database (SQLite)> Script

```
SELECT e.EmployeeId, e.FirstName as Employee, c.FirstName as Customer
from Employee as e
inner join Customer c
on e.EmployeeId = c.CustomerId
WHERE SupportRepId IN(SELECT EmployeeId FROM Employee)
```

Results 31 Results 32 Results 33 Results 34 InvoiceLine 35 Employee(+) 36

SELECT e.EmployeeId, e.FirstName as Employee, c.FirstName as Customer

	EmployeeId	Employee	Customer
1	1	Andrew	Luís
2	3	Jane	François
3	4	Margaret	Björn
4	5	Steve	František
5	8	Laura	Daan
6	2	Nancy	Leonie
7	6	Michael	Helena
8	7	Robert	Astrid

11. Show employee ID that never handles customers. (Use one of the INTERSECT/EXCEPT/UNION operators).

Answer:

```
SELECT EmployeeId
FROM Employee
INTERSECT
SELECT CustomerId
FROM Customer
WHERE SupportRepId NOT IN (SELECT EmployeeId FROM Employee)
```

The screenshot shows the DBeaver SQL editor with the following query:

```
SELECT EmployeeId
FROM Employee
INTERSECT
SELECT CustomerId
FROM Customer
WHERE SupportRepId NOT IN (SELECT EmployeeId FROM Employee)
```

The results grid is empty, indicating that no employees were found who do not handle customers.

12. Show customer first name, invoice ID, invoice date, and previous invoice date.

Answer:

```
SELECT c.FirstName, i.InvoiceId, i.InvoiceDate,
lag(i.InvoiceDate, 1) OVER (ORDER BY i.InvoiceDate ASC) PreviousInvoiceDate
FROM Customer as c
Inner join Invoice as i
on c.CustomerId = i.InvoiceId
limit 10
```

The screenshot shows the DBeaver SQL editor with the following query:

```
SELECT c.FirstName, i.InvoiceId, i.InvoiceDate,
lag(i.InvoiceDate, 1) OVER (ORDER BY i.InvoiceDate ASC) PreviousInvoiceDate
FROM Customer as c
Inner join Invoice as i
on c.CustomerId = i.InvoiceId
limit 10
```

The results grid shows the following data:

	FirstName	InvoiceId	InvoiceDate	PreviousInvoiceDate
1	Luis	1	2007-01-01 00:00:00	[NULL]
2	Leonie	2	2007-01-02 00:00:00	2007-01-01 00:00:00
3	François	3	2007-01-03 00:00:00	2007-01-02 00:00:00
4	Björn	4	2007-01-06 00:00:00	2007-01-03 00:00:00
5	František	5	2007-01-11 00:00:00	2007-01-06 00:00:00
6	Helena	6	2007-01-19 00:00:00	2007-01-11 00:00:00
7	Astrid	7	2007-02-01 00:00:00	2007-01-19 00:00:00
8	Daan	8	2007-02-01 00:00:00	2007-02-01 00:00:00
9	Kara	9	2007-02-02 00:00:00	2007-02-01 00:00:00
10	Eduardo	10	2007-02-03 00:00:00	2007-02-02 00:00:00

13. Show the customer's country origin, number of invoices, and country category based on the following criteria → 'Success' if the country has more than 20 invoices, and 'Need Improvement' if the country has less than 20 invoices. (Can use Subquery or CTE).

Expected output (partial data, not whole data):

	Country	InvoiceCount	CountryCategory
1	Argentina	7	Need Improvement
2	Australia	7	Need Improvement
3	Austria	7	Need Improvement
4	Belgium	7	Need Improvement
5	Brazil	35	Success
6	Canada	56	Success

Answer:

```

with invoicecountry as(
select BillingCountry as Country, count(BillingCountry) as InvoiceCount
from Invoice i
group by (BillingCountry))

select Country, InvoiceCount,
case when InvoiceCount > 20 then 'Success' WHEN InvoiceCount < 20 then 'Need Improvement'
END 'CountryCategory'
from invoicecountry
limit 10

```

Invoice 32 Invoice 33 Invoice 34 Invoice 35 Results 36 Invoice 37 Invoice 38 Invoice 39

with invoicecountry as(select Billing

	Country	InvoiceCount	CountryCategory
1	Argentina	7	Need Improvement
2	Australia	7	Need Improvement
3	Austria	7	Need Improvement
4	Belgium	7	Need Improvement
5	Brazil	35	Success
6	Canada	56	Success
7	Chile	7	Need Improvement
8	Czech Republic	14	Need Improvement
9	Denmark	7	Need Improvement
10	Finland	7	Need Improvement