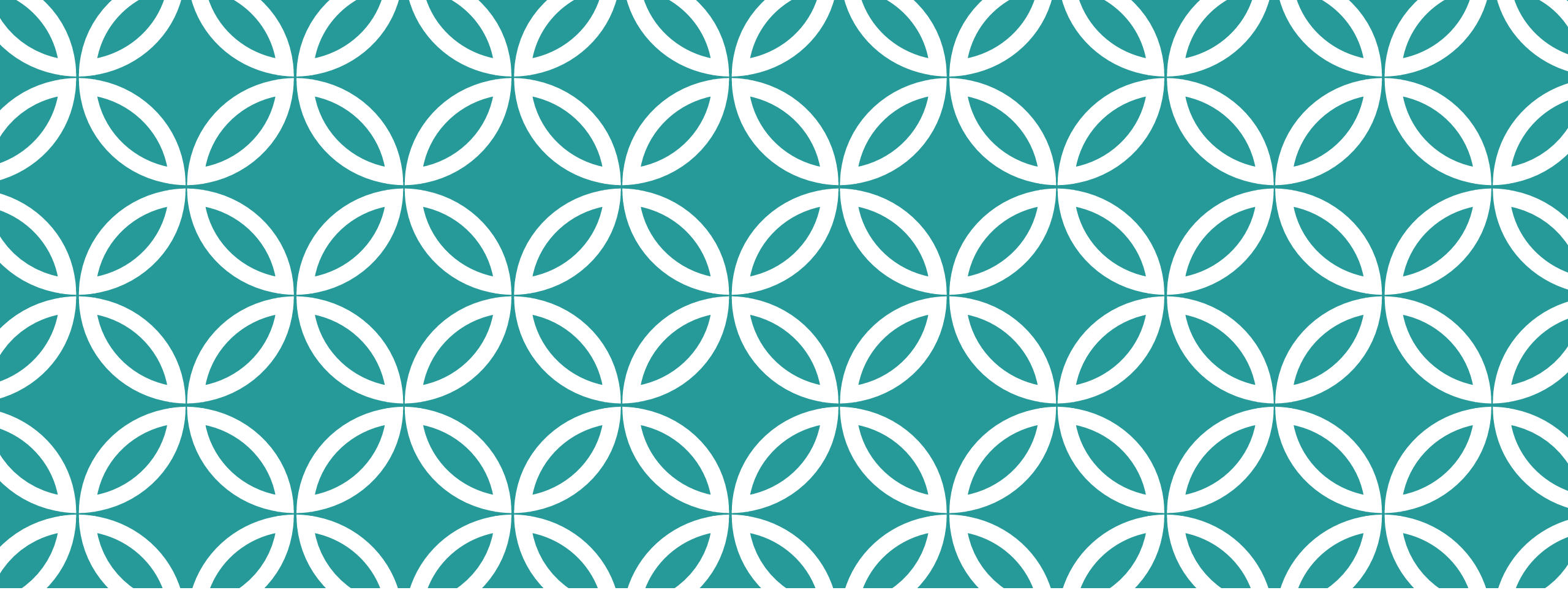


CHAPTER 12: SUBQUERIES

2023

CHAPTER 12: SUBQUERIES

Content:

1. Introduction to subqueries
2. subquery guidelines
3. Subquery rules
4. textual subqueries
5. CTEs

ATTEMPT THE FOLLOWING:

1. Show all listings that are above avg in price

2. Show the price and the names of the most expensive listings (for example if the highest listing = R250 000 and there are 3 such listings I want to see all 3. This can vary each day)

1. Show all listings that are above avg in price

Select name from listings

Where price > (select avg(price) from listings)

This inner query will result in 1 answer

This inner query will appear in brackets

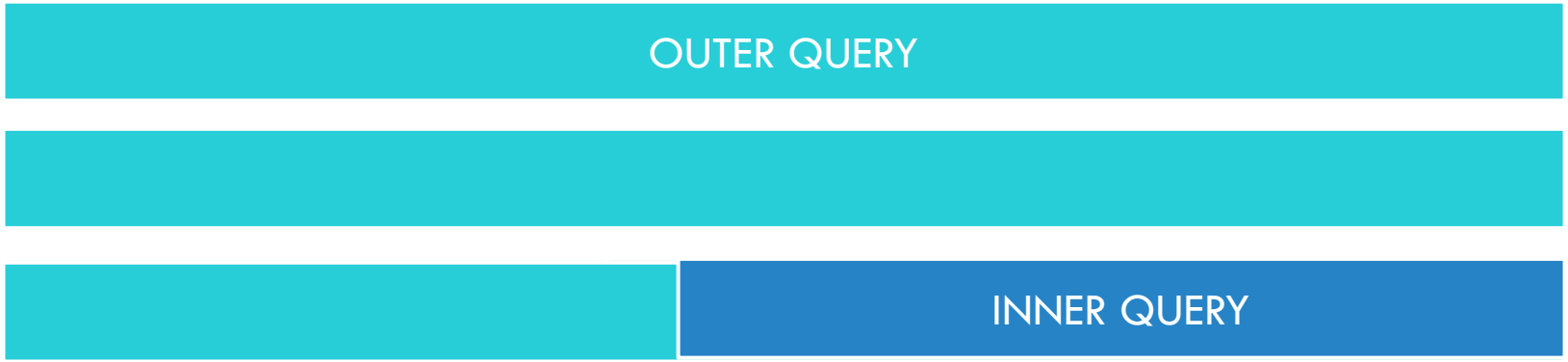
2. Show the price and the names of the most expensive listings (for example if the highest listing = R250 000 and there are 3 such listings I want to see all 3.)

Select name, price from listings

Where price = (select max(price) from listings)

SUBQUERY

To accomplish what we need to with 2 queries we can have a subquery in a main query



SUBQUERY

A subquery is a query that is nested inside a SELECT, INSERT , UPDATE or DELETE statement or inside another subquery.

A subquery may occur in :

- A SELECT clause
- A FROM clause
- A WHERE clause

(examples will follow)

You can use the comparison operators such as > < or =. This is used when the inner query results in one answer

A subquery nested in the outer SELECT statement has the following components:

- A regular SELECT query including the regular select list components.
- A regular FROM clause including one or more table or view names.
- An optional WHERE clause.
- An optional GROUP BY clause.
- An optional HAVING clause.

SOME GUIDELINES -SUBQUERIES

A subquery must be enclosed in → (parentheses)

A subquery must be placed on the right side of the comparison operator.

ORDER BY clause cannot be added into a subquery.

- You can use an ORDER BY clause in the main SELECT statement (outer query) which will be the last clause.

If a subquery (inner query) returns a null value to the outer query, the outer query will not return any rows when using certain comparison operators in a WHERE clause.

3 different scenarios how we'll use subqueries:

SELECT

Select school_name, school_district, passrate, (select max(passrate) from tblResults)

from school_data

School_name	School_district	Passrate	max(passrate)
ROYAL KINGS SCHOOL-MIDRAND	JOHANNESBURG EAST	97,2	98,5
DIEPSLOOT EAST PRIMARY SCHOOL	JOHANNESBURG NORTH	95	98,5
THINKINC ACADEMY	EKURHULENI NORTH	98,5	98,5
OLYMPUS EDUCATIONAL INSTITUTE	JOHANNESBURG EAST	96,8	98,5
THE MUSTARD SEED PRE PRIMARY AND PRIMARY SCHOOL	JOHANNESBURG SOUTH	92	98,5

FROM

```
Select country, avg(num_orders) from
(
    Select custID, Country, count (*) as num_orders
    From orders
    Group by 1,2
    ) as subQ
Group by country
```

WHERE

Select * from orders

where empID In (Select empID from Employees where surname like ‘van%’)

APPLICANTS & POSITIONSAVAILABLE TABLES

Applicants				
Fname	**L_Name**	**Job_Title**	**ID**	**Cell**
Peter	Parker	Analyst	432	2774
Susan	Smith	Teacher	655	2784
Jack	Johnson	Analyst	543	2783
Philip	Lee	Economist	754	2784

PositionsAvailable	
Title	**Company**
Analyst	Spar
Client Care	Absa
Economist	Absa
Assistant	PnP
Manager	PL
Manager	IG

Two tables about people applying for positions, and actual available positions

Show all details of only the applicants who match any of the positions on the PositionAvailable table

Hint – you can use the “in” function together with a subquery

*Select * from Applicants*

Where job_title in (select title from PositionAvailable)

This inner query will result in more than one answer – therefore we use “in”

Find the name of the company that placed order 10290:

```
SELECT CompanyName
FROM Customers
WHERE CustomerID = (SELECT CustomerID FROM Orders WHERE OrderID = 10290);
```

SUBQUERY: CREATE TABLE

```
CREATE TABLE new_table
  AS (SELECT * FROM old_table);


CREATE TABLE suppliers
  AS (SELECT id,  address,  city,  state,  zip FROM companies
      WHERE id > 1000);
```

SUBQUERY– EXPLAIN ONLY

1.

```
select title,  char_length(title)
from books
where char_length(title)  = (select max(char_length(title))from books) ;
```

2.

```
SELECT Name FROM Product
WHERE ProductSubcategoryID IN
    (SELECT ProductSubcategoryID FROM
ProductSubcategory
WHERE [Name] =  Wheels );
```

3.

```
SELECT [Name]
FROM Production WHERE ListPrice >
    (SELECT AVG (ListPrice) FROM Production);
```

4.

```
SELECT * FROM crime_incidents_2014
 WHERE Date = (SELECT MIN(date) FROM
crime_incidents_2014)
```

5.

```
SELECT * FROM customers
WHERE id IN (SELECT DISTINCT customer_id FROM orders WHERE
cost > 200);
```

6.

```
SELECT employee_id,  first_name,  last_name,  salary
FROM    employees
WHERE salary = (SELECT MAX(salary) FROM employees)
ORDER BY first_name    last_name;
```

7.

```
SELECT employee_id,  first_name,  last_name,  salary
FROM employees
WHERE salary > (SELECT AVG(salary) FROM employees);
```

SUBQUERY RULES

- The select list of a subquery introduced with a comparison operator can include only **one** expression or column name (except when using Exists / Any / all / some / In).
- Because they must return a single value subqueries cannot include GROUP BY and HAVING clauses.
- The DISTINCT keyword cannot be used with subqueries that include `GROUP BY.

IN ANY EXISTS SOME or ALL can be used where the inner query results in multiple answers

SUBQUERY: SOME / ALL / EXIST / ANY

Subqueries can be composed using the following commands:

- SOME- some of data satisfy the query selection
- ALL- all of data should satisfy the query selection
- EXIST- there exists data that satisfies the query selection
- ANY- any of the data satisfies the query selection

EXISTS

The SQL EXISTS Operator

The EXISTS operator is used to test for the existence of any record in a subquery.

The EXISTS operator returns TRUE if the subquery returns one or more records.

EXISTS

```
SELECT column_name(s)
FROM table_name
WHERE EXISTS
(SELECT column_name FROM table_name WHERE condition);
```

```
SELECT SupplierName
FROM Suppliers
WHERE EXISTS (SELECT ProductName FROM Products
WHERE Products.SupplierID = Suppliers.supplierID AND Price < 20);
```

ANY

The SQL ANY Operator

The ANY operator:

returns a Boolean value as a result

returns TRUE if ANY of the subquery values meet the condition

ANY means that the condition will be true if the operation is true for any of the values in the range.

ANY

```
SELECT column_name(s)
FROM table_name
WHERE column_name operator ANY
(SELECT column_name FROM table_name WHERE condition);
```

```
SELECT ProductName
FROM Products
WHERE ProductID = ANY (SELECT ProductID FROM OrderDetails WHERE Quantity
= 10);
```

```
SELECT [Name] FROM Product
WHERE ProductSubcategoryID
=ANY (SELECT ProductSubcategoryID FROM ProductSubcategory WHERE Name =
Wheels )
```

ALL

The ALL operator:

- returns a Boolean value as a result
- returns TRUE if ALL of the subquery values meet the condition
- is used with SELECT, WHERE and HAVING statements
- ALL means that the condition will be true only if the operation is true for all values in the range.

ALL

```
SELECT column_name(s)
FROM table_name
WHERE column_name operator ALL
(SELECT column_name FROM table_name WHERE condition);
```

```
SELECT ProductName
FROM Products
WHERE ProductID = ALL (SELECT ProductID FROM OrderDetails WHERE Quantity
= 10);
```

Practice SUBQUERY

HR

Employee

employee_id	first_name	last_name	email	phone_number	hire_date	job_id	salary	manager_id	department_id
100	Steven	King	steven.king@sqltutorial.org	515.123.4567	1987/06/17	4	24000	0	9
101	Neena	Kochhar	neena.kochhar@sqltutorial.org	515.123.4568	1989/09/21	5	17000	100	9
102	Lex	De Haan	lex.de haan@sqltutorial.org	515.123.4569	1993/01/13	5	17000	100	9
103	Alexander	Hunold	alexander.hunold@sqltutorial.org	590.423.4567	1990/01/03	9	9000	102	6

Country

country_id	country_name	region_id
AR	Argentina	2
AU	Australia	3
BE	Belgium	1
BR	Brazil	2
CA	Canada	2

Dependant

dependent_id	first_name	last_name	relationship	employee_id
1	Penelope	Gietz	Child	206
2	Nick	Higgins	Child	205
3	Ed	Whalen	Child	200
4	Jennifer	King	Child	100
5	Johnny	Kochhar	Child	101

Region

region_id	region_name
1	Europe
2	Americas
3	Asia
4	Middle East and Africa

Department

department_id	department_name	location_id
1	Administration	1700
2	Marketing	1800
3	Purchasing	1700
4	Human Resources	2400

Jobs

job_id	job_title	min_salary	max_salary
1	Public Accountant	4200	9000
2	Accounting Manager	8200	16000
3	Administration Assistant	3000	6000

Location

location_id	street_address	postal_code	city	state_province	country_id
1400	2014 Jabberwocky Rd	26192	Southlake	Texas	US
1500	2011 Interiors Blvd	99236	South San Francisco	California	US
1700	2004 Charade Rd	98199	Seattle	Washington	US

Practice SUBQUERY

1. *Display the first and last names, of all the A**ccountant**s*
2. *Show all employee info of those who works in **Finance***
3. *How many employees get paid more than the average salary?*
4. *Only show the employees who wont be able to earn less than 8,000*

Practice SUBQUERY

1. Find all listing names with the maximum price
2. Find all listing host names where last review date is the first day review.
3. Find all listings with prices below the average price

CTE (WITH… AS …)

CTE stands for **Common Table Expression**.

A Named temporary result – it only exist from the moment you run this query until it's done running. No Temp file exists anywhere. It's a table being created from which when can then query from.

It's available temporarily in the execution scope of a statement such as SELECT, INSERT, UPDATE, DELETE, or MERGE.

The syntax: WITH expression_name AS

(CTE_definition)

SQL_statement;

CTE EXAMPLE

The values in brackets → to rename those 3 columns

```sql
WITH cte_sales_amounts (staff, sales, year) AS (
    SELECT
        first_name + ' ' + last_name,
        SUM(quantity * list_price * (1 - discount)),
        YEAR(order_date)
    FROM
        sales.orders o
    INNER JOIN sales.order_items i ON i.order_id = o.order_id
    INNER JOIN sales.staffs s ON s.staff_id = o.staff_id
    GROUP BY
        first_name + ' ' + last_name,
        year(order_date)
)

SELECT
    staff,
    sales
FROM
    cte_sales_amounts
WHERE
    year = 2018;
```

100 %

Results | Messages

	staff	sales
1	Genna Serrano	247174.3531
2	Mireya Copeland	230246.9328
3	Kali Vargas	135113.1647
4	Marcelene Boyer	520105.6064
5	Venita Daniel	625358.3947
6	Layla Terrell	56531.3358

Query executed successfully. | PLJHBTRALPT017\SQLEXPRESS (... | PLATINUMLIFE\Masego.Di... | master | 00:00:00 | 6 rows

ANOTHER EXAMPLE

```
WITH CTE_Employee as
(SELECT FirstName, LastName, Gender, Salary
, COUNT(gender) OVER (PARTITION by Gender) as TotalGender
, AVG(Salary) OVER (PARTITION BY Gender) as AvgSalary
FROM SQLTutorial..EmployeeDemographics emp
JOIN SQLTutorial..EmployeeSalary sal
    ON emp.EmployeeID = sal.EmployeeID
WHERE Salary > '45000'
)
Select *
FROM CTE_Employee
```

SUBQUERY VS CTE(WITH)

```sql
-- We want the average number of orders per customer per country
SELECT shipcountry, AVG(num_orders) FROM
    (SELECT customerid, shipcountry, count(*) AS num_orders
    FROM orders
    GROUP BY 1,2) sub
GROUP BY 1
```

```sql
-- We want the average number of orders per customer per country
WITH cte_orders AS (
    SELECT customerid, shipcountry, count(*) AS num_orders
    FROM orders
    GROUP BY 1,2)


SELECT shipcountry, AVG(num_orders)
FROM cte_orders
GROUP BY 1
```

Practice CTE

HR

Employee

employee_id	first_name	last_name	email	phone_number	hire_date	job_id	salary	manager_id	department_id
100	Steven	King	steven.king@sqltutorial.org	515.123.4567	1987/06/17	4	24000	0	9
101	Neena	Kochhar	neena.kochhar@sqltutorial.org	515.123.4568	1989/09/21	5	17000	100	9
102	Lex	De Haan	lex.de haan@sqltutorial.org	515.123.4569	1993/01/13	5	17000	100	9
103	Alexander	Hunold	alexander.hunold@sqltutorial.org	590.423.4567	1990/01/03	9	9000	102	6

Country

country_id	country_name	region_id
AR	Argentina	2
AU	Australia	3
BE	Belgium	1
BR	Brazil	2
CA	Canada	2

Dependant

dependent_id	first_name	last_name	relationship	employee_id
1	Penelope	Gietz	Child	206
2	Nick	Higgins	Child	205
3	Ed	Whalen	Child	200
4	Jennifer	King	Child	100
5	Johnny	Kochhar	Child	101

Region

region_id	region_name
1	Europe
2	Americas
3	Asia
4	Middle East and Africa

Department

department_id	department_name	location_id
1	Administration	1700
2	Marketing	1800
3	Purchasing	1700
4	Human Resources	2400

Jobs

job_id	job_title	min_salary	max_salary
1	Public Accountant	4200	9000
2	Accounting Manager	8200	16000
3	Administration Assistant	3000	6000

Location

location_id	street_address	postal_code	city	state_province	country_id
1400	2014 Jabberwocky Rd	26192	Southlake	Texas	US
1500	2011 Interiors Blvd	99236	South San Francisco	California	US
1700	2004 Charade Rd	98199	Seattle	Washington	US

Practice CTE

1. *How many employees work in each department*
2. *What is the total amount for Salaries, by Department and by Job Title?*