

Dokumentasi menampilkan data dalam tabel dalam Oracle Apex menggunakan query function, join, sub query, bind variable, view, conditional expression dan sequences.

1. Function

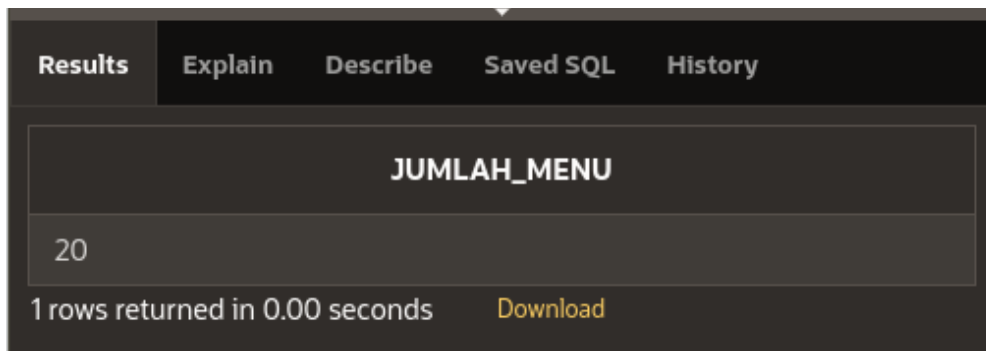
Contoh menggunakan function COUNT, SUM, MIN, MAX :

a. COUNT

Menggunakan function ini untuk menghitung jumlah menu yang ada di warung makan, dengan query :

```
SELECT COUNT(menu_name) AS jumlah_menu  
FROM menu;
```

Dengan hasil tampilan :



The screenshot shows the Oracle Apex interface with the 'Results' tab selected. The query result is displayed in a table with one column named 'JUMLAH_MENU' and one row containing the value '20'. Below the table, it indicates '1 rows returned in 0.00 seconds' and provides a 'Download' link.

JUMLAH_MENU
20

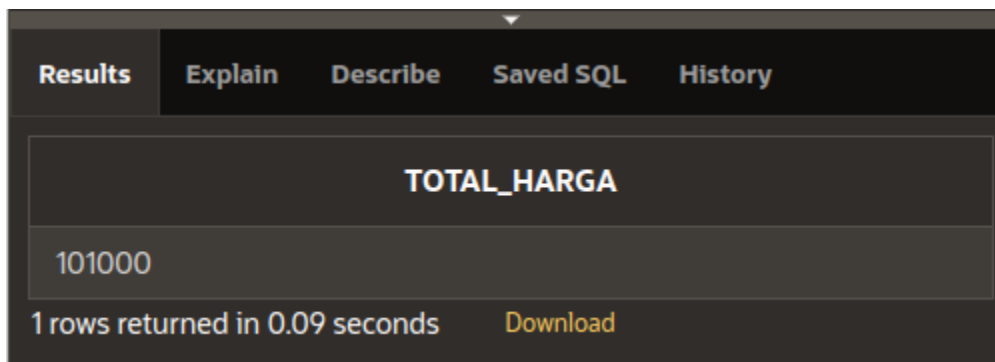
1 rows returned in 0.00 seconds [Download](#)

b. SUM

Menggunakan function ini untuk menghitung total harga semua menu yang ada di warung makan, dengan query :

```
SELECT SUM(price) AS total_harga  
FROM menu_price;
```

Dengan hasil tampilan :



The screenshot shows the Oracle Apex interface with the 'Results' tab selected. The query result is displayed in a table with one column named 'TOTAL_HARGA' and one row containing the value '101000'. Below the table, it indicates '1 rows returned in 0.09 seconds' and provides a 'Download' link.

TOTAL_HARGA
101000

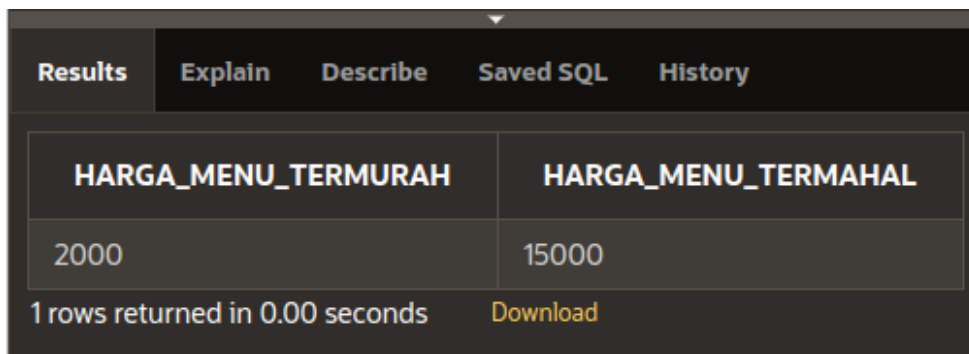
1 rows returned in 0.09 seconds [Download](#)

c. MIN dan MAX

Menggunakan function MIN untuk mencari harga menu termurah dan MAX untuk mencari harga menu termahal, dengan query :

```
SELECT MIN(price) AS harga_menu_termurah, MAX(price) AS  
harga_menu_termahal  
FROM menu_price;
```

Dengan hasil tampilan :



The screenshot shows a database query results window with a dark theme. At the top, there are tabs: 'Results' (selected), 'Explain', 'Describe', 'Saved SQL', and 'History'. Below the tabs is a table with two columns: 'HARGA_MENU_TERMURAH' and 'HARGA_MENU_TERMAHAL'. The first row of data shows the values '2000' and '15000' respectively. At the bottom of the results area, it says '1 rows returned in 0.00 seconds' and there is a 'Download' link.

HARGA_MENU_TERMURAH	HARGA_MENU_TERMAHAL
2000	15000

1 rows returned in 0.00 seconds [Download](#)

2. Join

Contoh penggunaan Join :

- Menggunakan JOIN untuk menggabungkan tabel menu dan menu_price, untuk bisa melihat menu_name dan pricenya, dengan query :

```
SELECT m.menu_name, mp.price  
FROM menu_price mp  
JOIN menu m ON mp.menu_id = m.menu_id;
```

Dengan hasil tampilan :

Results	Explain	Describe	Saved SQL	History
MENU_NAME		PRICE		
Nasi Goreng		12000		
Telur Balado		5000		
Indomie Goreng		8000		
Indomie Goreng Telur		10000		
Sayur Sop		4000		
Telur Rebus		3000		
Kopi Kapal Api		4000		
Nasi Putih		4000		
Nasi Goreng Spesial		15000		
Telur Dadar		3000		
Es Ovaltine		4000		

- b. Menggunakan JOIN untuk menggabungkan 3 tabel, antara invoice, customer, dan eatery_table, dengan query :

```
SELECT i.invoice_date, c.customer_name, t.eatery_table_name,
i.food_choice_id
```

```
FROM invoice i
```

```
JOIN customer c ON i.customer_id = c.customer_id
```

```
JOIN eatery_table t ON i.eatery_table_id = t.eatery_table_id;
```

Dengan hasil tampilan :

Results	Explain	Describe	Saved SQL	History
INVOICE_DATE	CUSTOMER_NAME	EATERY_TABLE_NAME	FOOD_CHOICE_ID	
01-Dec-2023	Dimas	01	EI	
02-Dec-2023	Ignatius	01	EI	
01-Dec-2023	William	01	EI	
01-Dec-2023	Ignatius	02	EI	
03-Dec-2023	Dimas	02	EI	

3. Sub Query

Menggunakan SUBQUERY dan JOIN untuk mendapatkan menu_name dari menu dan price dari menu_price, yang dimana kondisi price harus lebih besar sama dengan price yang minimum lebih besar dari 4000, dengan query :

```
SELECT m.menu_name, mp.price  
FROM menu_price mp  
JOIN menu m ON mp.menu_id = m.menu_id  
WHERE mp.price >= (  
    SELECT MIN(price)  
    FROM menu_price  
    WHERE price > 4000  
);
```

Dengan hasil tampilan :

Results

Explain

Describe

Saved SQL

History

MENU_NAME	PRICE
Nasi Goreng	12000
Telur Balado	5000
Indomie Goreng	8000
Indomie Goreng Telur	10000
Nasi Goreng Spesial	15000

5 rows returned in 0.01 seconds

Download

4. Bind Variable

- Menggunakan konsep ini untuk mencari customer_name pada tabel customer menggunakan WHERE nama_kolom = :alias, dengan query :

```
SELECT *  
FROM customer  
WHERE customer_name = :customer_name;
```

Dengan hasil tampilan :

The image shows two screenshots from an Oracle SQL Developer interface. The top screenshot is a window titled "Enter Bind Variables - Google ...". It has a URL bar showing "iacademy2.oracle.com/ords/f?p=4500:138:33...". Below the URL bar is a green "Submit" button. A table with two columns, "Bind Variable" and "Value", is displayed. The first row shows ":CUSTOMER_NAME" in the "Bind Variable" column and "Dimas" in the "Value" column. The bottom screenshot shows a "Results" window with tabs for "Results", "Explain", "Describe", "Saved SQL", and "History". The "Results" tab is active, showing a table with three columns: "CUSTOMER_ID", "CUSTOMER_NAME", and "PHONE_NUMBER". The first row of data shows "1" for "CUSTOMER_ID", "Dimas" for "CUSTOMER_NAME", and "089567891010" for "PHONE_NUMBER". Below the table, it says "1 rows returned in 0.02 seconds" and there is a "Download" link.

Bind Variable	Value
:CUSTOMER_NAME	Dimas

CUSTOMER_ID	CUSTOMER_NAME	PHONE_NUMBER
1	Dimas	089567891010

- b. Mencari customer_id pada tabel customer, dengan query sama seperti kita mencari customer_name :

SELECT *

FROM customer

WHERE customer_id = :customer_id;

Dengan hasil tampilan :

The screenshot shows a web browser window with the title "Enter Bind Variables - Google ...". The address bar displays the URL "iacademy2.oracle.com/ords/f?p=4500:138:33...". A green "Submit" button is located in the top right corner. Below the button is a table with two columns: "Bind Variable" and "Value". The first row shows the bind variable ":CUSTOMER_ID" with the value "1" entered in a text box. Below this table is a large empty space. At the bottom of the screenshot, there is a "Results" section with tabs for "Results", "Explain", "Describe", "Saved SQL", and "History". The "Results" tab is selected, showing a table with three columns: "CUSTOMER_ID", "CUSTOMER_NAME", and "PHONE_NUMBER". The first row contains the values "1", "Dimas", and "089567891010". Below the table, it says "1 rows returned in 0.02 seconds" and there is a "Download" link.

Bind Variable	Value
:CUSTOMER_ID	1

CUSTOMER_ID	CUSTOMER_NAME	PHONE_NUMBER
1	Dimas	089567891010

1 rows returned in 0.02 seconds [Download](#)

5. View

Menggunakan CREATE VIEW untuk menampilkan total penjualan per hari dan juga menggunakan function SUM, JOIN, dan GROUP BY untuk mengelompokkan berdasarkan invoice_date. Dengan query :

CREATE VIEW total_penjualan AS

SELECT i.invoice_date, SUM(d.quantity * mp.price) AS total_penjualan_perhari

FROM invoice_detail d

JOIN menu_price mp ON d.menu_price_id = mp.menu_price_id

JOIN invoice i ON i.invoice_id = d.invoice_id

GROUP BY i.invoice_date;

Setelah membuat view, bisa kita tampilkan hasilnya menggunakan query :

SELECT * FROM total_penjualan;

Dengan hasil tampilan :

Results

Explain

Describe

Saved SQL

History

INVOICE_DATE	TOTAL_PENJUALAN_PERHARI
01-Dec-2023	36000
02-Dec-2023	25000
03-Dec-2023	25000

3 rows returned in 0.32 seconds

Download

6. Conditional Expression

- Menggunakan TO_CHAR, CASE, WHEN, THEN, ELSE untuk membuat kondisi jika bulan invoice_date sesuai dengan format bulan yang ditulis dalam WHEN maka akan menjadi bulan dalam bahasa indonesia. Dengan query :

```
SELECT invoice_date,  
CASE TO_CHAR(invoice_date, 'Mon')  
WHEN 'Jan' THEN 'Januari'  
WHEN 'Feb' THEN 'Februari'  
WHEN 'Mar' THEN 'Maret'  
WHEN 'Apr' THEN 'April'  
WHEN 'May' THEN 'Mei'  
WHEN 'Jun' THEN 'Juni'  
WHEN 'Jul' THEN 'Juli'  
WHEN 'Aug' THEN 'Agustus'  
WHEN 'Sep' THEN 'September'  
WHEN 'Oct' THEN 'Oktober'  
WHEN 'Nov' THEN 'November'  
ELSE 'Desember'  
END AS month_name
```

FROM invoice;

Dengan hasil tampilan :

Results	Explain	Describe	Saved SQL	History
INVOICE_DATE	MONTH_NAME			
03-Dec-2023	Desember			
03-Dec-2023	Desember			
03-Dec-2023	Desember			
01-Dec-2023	Desember			
02-Dec-2023	Desember			

- b. Contoh lain untuk menambahkan deskripsi price menu di antara low, average, high atau other. Dengan query :

```

SELECT m.menu_name, mp.price,
CASE
WHEN price > 6000 THEN 'High Price'
WHEN price > 3000 THEN 'Average Price'
WHEN price > 1000 THEN 'Low Price'
ELSE 'Other'
END AS description_price
FROM menu_price mp JOIN menu m ON mp.menu_id = m.menu_id;
    
```

Dengan hasil tampilan :

Results	Explain	Describe	Saved SQL	History
MENU_NAME	PRICE	DESCRIPTION_PRICE		
Nasi Goreng	12000	High Price		
Telur Balado	5000	Average Price		
Indomie Goreng	8000	High Price		
Indomie Goreng Telur	10000	High Price		
Sayur Sop	4000	Average Price		

7. Sequence

Contoh penggunaan sequence :

```
CREATE SEQUENCE customer_seq  
MINVALUE 1  
MAXVALUE 10000  
START WITH 5  
INCREMENT BY 1;
```

Penjelasan kode :

- CREATE SEQUENCE customer_seq: Memulai pembuatan sebuah urutan (sequence) dengan nama "customer_seq"
- MINVALUE 1: Menentukan nilai minimum untuk urutan. Dalam hal ini, nilai minimumnya adalah 1. Ini berarti urutan tidak akan pernah kurang dari nilai ini
- MAXVALUE 10000: Menentukan nilai maksimum untuk urutan. Dalam hal ini, nilai maksimumnya adalah 10000. Ini berarti urutan tidak akan pernah melebihi nilai ini
- START WITH 5: Menentukan nilai awal untuk urutan. Dalam hal ini, urutan akan dimulai dari nilai 5
- INCREMENT BY 1: Menentukan jumlah penambahan (increment) untuk setiap nilai dalam urutan. Dalam hal ini, setiap kali urutan digunakan, nilainya akan bertambah sebanyak 1

Setelah membuat sequence, bisa menambah/INSERT data dalam tabel customer dengan customer_id menggunakan sequence, dengan query :

```
INSERT INTO customer  
VALUES (customer_seq.NEXTVAL, 'Hadi', '089567891014');
```

Maka ketika kita **SELECT * FROM customer;**, akan menampilkan :

Results	Explain	Describe	Saved SQL	History
CUSTOMER_ID	CUSTOMER_NAME	PHONE_NUMBER		
1	Dimas	089567891010		
2	Ignatius	089567891011		
5	Hadi	089567891014		
3	William	089567891012		
4	Gabriel	089567891013		
5 rows returned in 0.00 seconds			Download	

Dengan INSERT menggunakan sequence, maka customer_id akan langsung terbuat.