

VERSION 2.0
FEBRUARI, 2020



DESAIN DATABASE

MODUL 4: SUB QUERY, VIEW

TIM PENYUSUN: - AGUS EKO MINARNO, S. KOM, M. KOM
- BYAS AGENG HAQIQIE

PRESENTED BY: LABORATORIUM INFORMATIKA
UNIVERSITAS MUHAMMADIYAH MALANG

DESAIN DATABASE

CAPAIAN PEMBELAJARAN MATA KULIAH

Diharapkan setelah mengerjakan modul ini, praktikan mampu untuk memahami klausa sub query, view

SUB CAPAIAN PEMBELAJARAN MATA KULIAH

Praktikan mampu mengimplementasikan subquery, view

KEBUTUHAN HARDWARE & SOFTWARE

- Oracle XE
- HR Scheme

MATERI POKOK

- Subquery
- View

MATERI PRAKTIKUM

SUB QUERY/NESTED QUERY

Subquery, inner query, atau nested query adalah sebuah query yang berada di dalam query SQL lain. Subquery biasanya di gunakan untuk mengembalikan/*return* data yang akan digunakan pada main query sebagai sebuah kondisi data yang akan di tampilkan nantinya.

Subquery dapat digunakan pada statement SELECT, INSERT, UPDATE, dan DELETE bersamaan dengan operator-operator di dalam query, seperti =, <=, <, >, >=, <=>, IN, BETWEEN, dan operator lainnya.

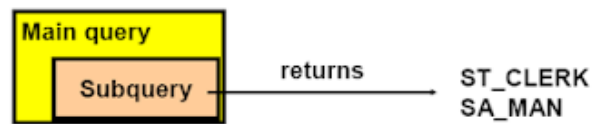
Ada beberapa aturan di dalam membuat subquery, yaitu :

- Subquery diletakkan di dalam tanda kurung.
- Operator ORDER BY tidak dapat digunakan di dalam subquery.
- Subquery hanya boleh memiliki satu kolom.
- Gunakan operator single-row dengan single-row subqueries, dan gunakan operator multiple-row dengan multiple-row subqueries. Single-row subqueries merupakan query yang hanya mengembalikan satu record (baris) dari inner statement SELECT. Multiple-row subqueries merupakan query yang mengembalikan lebih dari satu record (baris) dari inner statement SELECT.

- **Single-row subquery**



- **Multiple-row subquery**



- Operator BETWEEN tidak dapat digunakan dengan subquery. Namun, operator BETWEEN dapat digunakan dalam subquery.

Berikut merupakan contoh penggunaan subquery.

Penggunaan subquery dengan statement SELECT

Berikut merupakan struktur dasar dari subquery nya :

```
SELECT column_name [, column_name ]
FROM   table1 [, table2 ]
WHERE  column_name OPERATOR
      (SELECT column_name [, column_name ]
FROM   table1 [, table2 ]
[WHERE])
```

Contoh penggunaannya pada skema HR :

```
SELECT *
FROM EMPLOYEES
WHERE EMPLOYEE_ID IN (SELECT EMPLOYEE_ID
FROM EMPLOYEES
WHERE SALARY > 10000) ;
```

Dari query tersebut, maka akan menampilkan record seperti di bawah ini :

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT	MANAGER_ID	DEPARTMENT_ID
100	Steven	King	SKING	515.123.4567	06/17/2003	AD_PRES	24000	-	-	90
101	Neena	Kochhar	NKOCHHAR	515.123.4568	09/21/2005	AD_VP	17000	-	100	90
102	Lex	De Haan	LDEHAAN	515.123.4569	01/13/2001	AD_VP	17000	-	100	90
108	Nancy	Greenberg	NGREENBE	515.124.4569	08/17/2002	FI_MGR	12008	-	101	100
114	Den	Raphaely	DRAPHEAL	515.127.4561	12/07/2002	PU_MAN	11000	-	100	30
145	John	Russell	JRUSSEL	011.44.1344.429268	10/01/2004	SA_MAN	14000	.4	100	80
146	Karen	Partners	KPARTNER	011.44.1344.467268	01/05/2005	SA_MAN	13500	.3	100	80
147	Alberto	Errazuriz	AERRAZUR	011.44.1344.429278	03/10/2005	SA_MAN	12000	.3	100	80
148	Gerald	Cambrault	GCAMBRAU	011.44.1344.619268	10/15/2007	SA_MAN	11000	.3	100	80
149	Eleni	Zlotkey	EZLOTKEY	011.44.1344.429018	01/29/2008	SA_MAN	10500	.2	100	80

More than 10 rows available. Increase rows selector to view more rows.

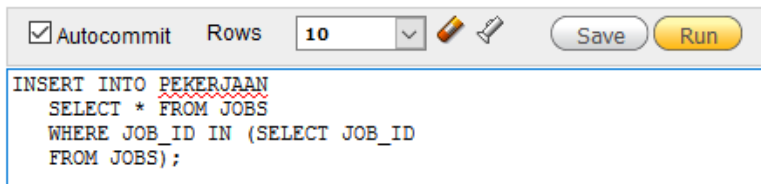
10 rows returned in 0.00 seconds [Download](#)

Penggunaan subquery dengan statement INSERT

Berikut merupakan struktur dasar dari subquery nya :

```
INSERT INTO table_name [ (column1 [, column2 ]) ]
  SELECT [ *|column1 [, column2 ]
  FROM table1 [, table2 ]
  [ WHERE VALUE OPERATOR ]
```

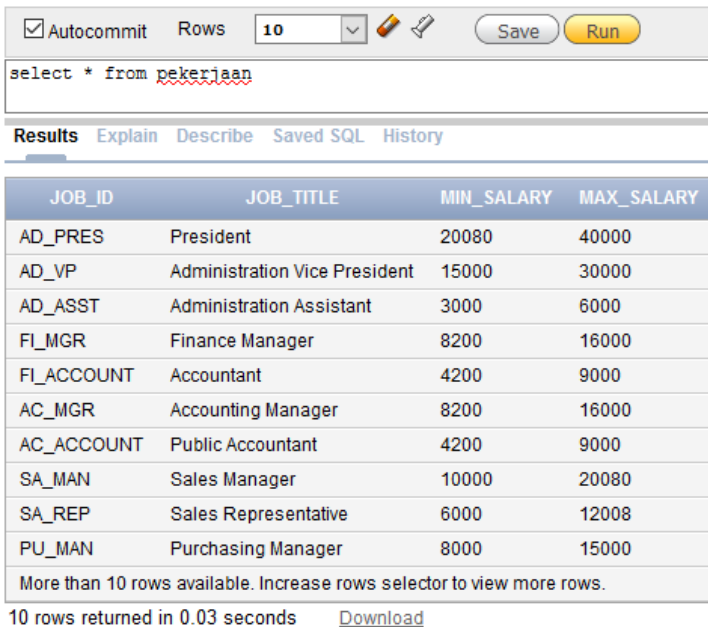
Contoh penggunaannya pada skema HR :



```
INSERT INTO PEKERJAAN
  SELECT * FROM JOBS
  WHERE JOB_ID IN (SELECT JOB_ID
  FROM JOBS);
```

Table pekerjaan merupakan table yang memiliki struktur yang sama persis dengan table jobs. Dan query di atas merupakan query untuk menyalin keseluruhan data yang ada di table jobs ke dalam table pekerjaan.

Dari query tersebut, maka akan menghasilkan record seperti di bawah ini :



select * from pekerjaan

Results Explain Describe Saved SQL History

JOB_ID	JOB_TITLE	MIN_SALARY	MAX_SALARY
AD_PRES	President	20080	40000
AD_VP	Administration Vice President	15000	30000
AD_ASST	Administration Assistant	3000	6000
FI_MGR	Finance Manager	8200	16000
FI_ACCOUNT	Accountant	4200	9000
AC_MGR	Accounting Manager	8200	16000
AC_ACCOUNT	Public Accountant	4200	9000
SA_MAN	Sales Manager	10000	20080
SA_REP	Sales Representative	6000	12008
PU_MAN	Purchasing Manager	8000	15000

More than 10 rows available. Increase rows selector to view more rows.

10 rows returned in 0.03 seconds [Download](#)

Penggunaan subquery dengan statement UPDATE

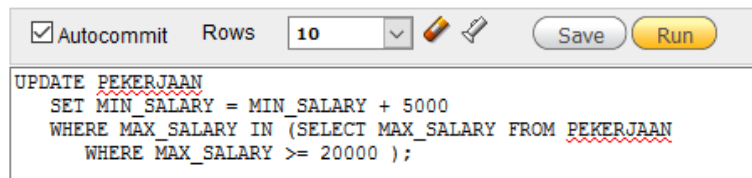
Berikut merupakan struktur dasar dari subquery nya :

```

UPDATE table
SET column_name = new_value
[ WHERE OPERATOR [ VALUE ]
    (SELECT COLUMN_NAME
    FROM TABLE_NAME)
[ WHERE) ]

```

Contoh penggunaannya pada skema HR :



Dari query tersebut, maka akan menghasilkan record seperti di bawah ini :

JOB_ID	JOB_TITLE	MIN_SALARY	MAX_SALARY
AD_PRES	President	25080	40000
AD_VP	Administration Vice President	20000	30000
AD_ASST	Administration Assistant	3000	6000
FI_MGR	Finance Manager	8200	16000
FI_ACCOUNT	Accountant	4200	9000
AC_MGR	Accounting Manager	8200	16000
AC_ACCOUNT	Public Accountant	4200	9000
SA_MAN	Sales Manager	15000	20080
SA_REP	Sales Representative	6000	12008
PU_MAN	Purchasing Manager	8000	15000
More than 10 rows available. Increase rows selector to view more rows.			

10 rows returned in 0.01 seconds [Download](#)

Penggunaan subquery dengan statement DELETE

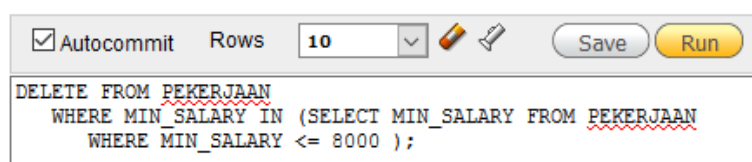
Berikut merupakan struktur dasar dari subquery nya :

```

DELETE FROM TABLE_NAME
[ WHERE OPERATOR [ VALUE ]
    (SELECT COLUMN_NAME
    FROM TABLE_NAME)
[ WHERE) ]

```

Contoh penggunaannya pada skema HR :



Dari query tersebut, maka akan menghasilkan record seperti di bawah ini :

JOB_ID	JOB_TITLE	MIN_SALARY	MAX_SALARY
AD_PRES	President	25080	40000
AD_VP	Administration Vice President	20000	30000
FI_MGR	Finance Manager	8200	16000
AC_MGR	Accounting Manager	8200	16000
SA_MAN	Sales Manager	15000	20080
MK_MAN	Marketing Manager	9000	15000

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

Contoh-contoh penggunaan subquery diatas merupakan multiple-row subqueries, di karenakan operator yag di gunakan merupakan operator multiple-row (IN). Untuk contoh penggunaan single-row subqueries dapat di lihat pada contoh di bawah ini :

```

SELECT CONCAT(CONCAT(E.FIRST_NAME, ' '), E.LAST_NAME) AS NAMA,
       E.HIRE_DATE AS "MULAI BEKERJA",
       J.JOB_TITLE AS PEKERJAAN,
       CONCAT(CONCAT(F.FIRST_NAME, ' '), F.LAST_NAME) AS "NAMA MANAGER"
FROM EMPLOYEES E
INNER JOIN JOBS J
    ON E.JOB_ID=J.JOB_ID
INNER JOIN EMPLOYEES F
    ON E.MANAGER_ID=F.EMPLOYEE_ID
WHERE E.MANAGER_ID = (SELECT DISTINCT A.MANAGER_ID FROM EMPLOYEES A WHERE A.MANAGER_ID = 103);

```

Dari query tersebut, maka akan menghasilkan record seperti di bawah ini :

NAMA	MULAI BEKERJA	PEKERJAAN	NAMA MANAGER
David Austin	06/25/2005	Programmer	Alexander Hunold
Valli Pataballa	02/05/2006	Programmer	Alexander Hunold
Diana Lorentz	02/07/2007	Programmer	Alexander Hunold
Bruce Ernst	05/21/2007	Programmer	Alexander Hunold

4 rows returned in 0.03 seconds [Download](#)

VIEW

View adalah sebuah table virtual hasil dari sebuah query statement SELECT. Mengapa view di sebut table virtual? Hal ini di karenakan view mirip sebuah table yang berisi kolom atau field dan record data tetapi view bersifat *read-only*, jadi kita tidak dapat melakukan proses insert dan sebagainya. Data yang ada di view akan selalu *up to date* seiring dengan data yang ada di table.

Membuat View

Untuk membuat sebuah view, kita dapat menggunakan perintah CREATE VIEW. Berikut merupakan query dasar untuk membuat view.

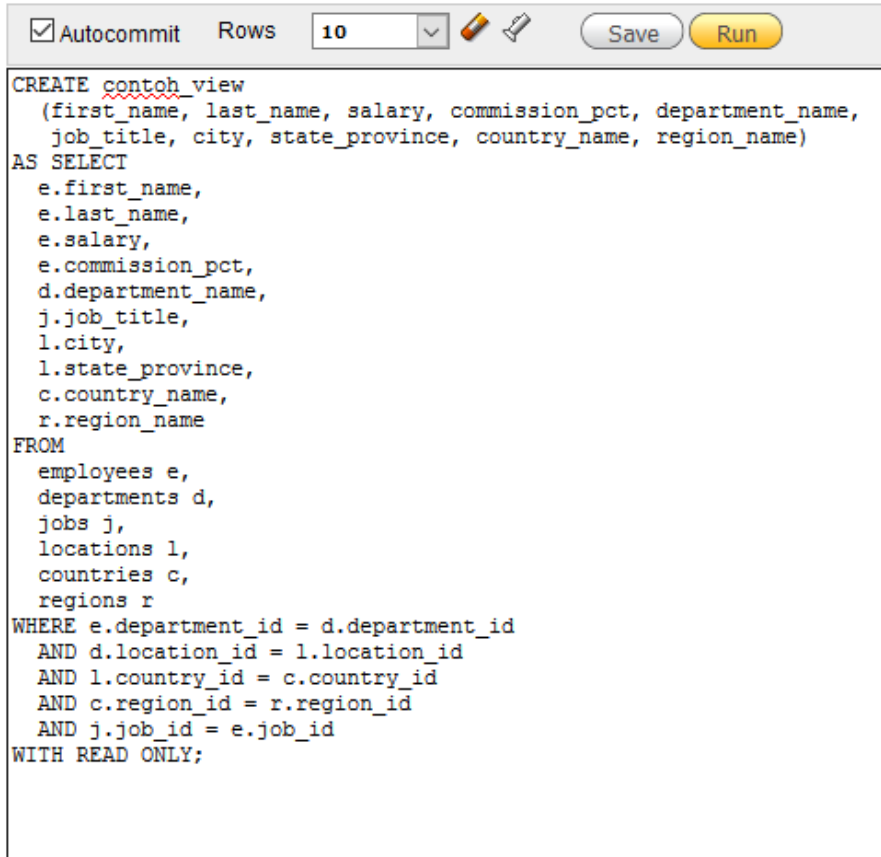
```

CREATE VIEW nama_view AS
SELECT kolom_1, kolom_2, kolom_n
FROM nama_table

```

```
WHERE kondisi;
```

Di bawah ini adalah contoh pembuatan view pada skema HR.



The screenshot shows a SQL IDE interface with a toolbar at the top containing 'Autocommit' (checked), 'Rows' (set to 10), and 'Save' and 'Run' buttons. The main text area contains the following SQL code:

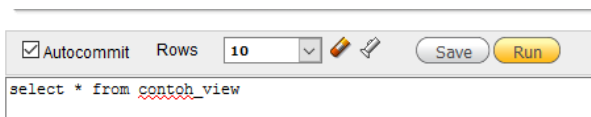
```
CREATE contoh_view
  (first_name, last_name, salary, commission_pct, department_name,
   job_title, city, state_province, country_name, region_name)
AS SELECT
  e.first_name,
  e.last_name,
  e.salary,
  e.commission_pct,
  d.department_name,
  j.job_title,
  l.city,
  l.state_province,
  c.country_name,
  r.region_name
FROM
  employees e,
  departments d,
  jobs j,
  locations l,
  countries c,
  regions r
WHERE e.department_id = d.department_id
      AND d.location_id = l.location_id
      AND l.country_id = c.country_id
      AND c.region_id = r.region_id
      AND j.job_id = e.job_id
WITH READ ONLY;
```

*pada query diatas, baris pertama seharusnya CREATE VIEW contoh_view.

Setelah menjalankan query diatas, maka view telah di buat. Untuk menampilkan view tersebut, dapat menggunakan statement SELECT.

```
SELECT * FROM VIEW_NAME;
```

Implementasinya adalah sebagai berikut.



The screenshot shows the same SQL IDE interface as before, but the text area now contains the following SQL query:

```
select * from contoh_view
```

View tersebut akan menampilkan data sebagai berikut.

FIRST_NAME	LAST_NAME	SALARY	COMMISSION_PCT	DEPARTMENT_NAME	JOB_TITLE	CITY	STATE_PROVINCE	COUNTRY_NAME	REGION_NAME
Steven	King	24000	-	Executive	President	Seattle	Washington	United States of America	Americas
Neena	Kochhar	17000	-	Executive	Administration Vice President	Seattle	Washington	United States of America	Americas
Lex	De Haan	17000	-	Executive	Administration Vice President	Seattle	Washington	United States of America	Americas
Alexander	Hunold	9000	-	IT	Programmer	Southlake	Texas	United States of America	Americas
Bruce	Ernst	6000	-	IT	Programmer	Southlake	Texas	United States of America	Americas
David	Austin	4800	-	IT	Programmer	Southlake	Texas	United States of America	Americas
Valli	Pataballa	4800	-	IT	Programmer	Southlake	Texas	United States of America	Americas
Diana	Lorentz	4200	-	IT	Programmer	Southlake	Texas	United States of America	Americas
Nancy	Greenberg	12008	-	Finance	Finance Manager	Seattle	Washington	United States of America	Americas
Daniel	Faviet	9000	-	Finance	Accountant	Seattle	Washington	United States of America	Americas
More than 10 rows available. Increase rows selector to view more rows.									
10 rows returned in 0.04 seconds Download									

Update View

Kita dapat meng-update view sesuai dengan kebutuhan, seperti menambahkan kolom pada view ataupun menambahkan filter pada klausa WHERE di view tersebut. Untuk mengupdate view, kita dapat menggunakan perintah CREATE OR REPLACE VIEW.

Berikut merupakan query dasar untuk mengupdate view.

```
CREATE OR REPLACE VIEW nama_view AS

SELECT kolom_1, kolom_2, kolom_n

FROM nama_tabel

WHERE kondisi;
```

Contoh implementasinya dapat dilihat di bawah ini.

```

Autocommit Rows 10 Save Run

CREATE OR REPLACE VIEW contoh_view
(first_name, last_name, salary, commission_pct, department_name,
job_title, city, state_province, country_name, region_name)
AS SELECT
e.first_name,
e.last_name,
e.salary,
e.commission_pct,
d.department_name,
j.job_title,
l.city,
l.state_province,
c.country_name,
r.region_name
FROM
employees e,
departments d,
jobs j,
locations l,
countries c,
regions r
WHERE e.department_id = d.department_id
AND d.location_id = l.location_id
AND l.country_id = c.country_id
AND c.region_id = r.region_id
AND j.job_id = e.job_id
AND j.job_id = 'AC_ACCOUNT'
WITH READ ONLY;
```

Query tersebut akan menghasilkan view seperti dibawah ini.

FIRST_NAME	LAST_NAME	SALARY	COMMISSION_PCT	DEPARTMENT_NAME	JOB_TITLE	CITY	STATE_PROVINCE	COUNTRY_NAME	REGION_NAME
William	Gietz	8300	-	Accounting	Public Accountant	Seattle	Washington	United States of America	Americas



1 rows returned in 0.14 seconds [Download](#)

Menghapus View

Berikut query untuk menghapus view yang telah di buat

```
DROP VIEW nama_view;
```

Contoh :

☒ Autocommit
 Rows



```
DROP VIEW contoh_view
```

LEMBAR KERJA

KEGIATAN 1

- Gunakan skema HR. Buatlah subquery yang akan menampilkan record data **job_title**, **department_name**, **name(first name dan last name di gabung)**, **hire_date** dan yang mulai bekerja antara tahun 1995 hingga 2007 seperti di bawah ini.

NAMA_PKERJAAN	DEPARTEMEN	NAMA	MULAI_BKERJA
Programmer	IT	LexDe Haan	01/13/2001
Public Accountant	Accounting	NeenaKochhar	09/21/1997
Accounting Manager	Accounting	NeenaKochhar	10/28/2001
Marketing Representative	Marketing	MichaelHartstein	02/17/2004
Stock Clerk	Shipping	DenRaphaely	03/24/2006
Stock Clerk	Shipping	PayamKaufling	01/01/2007
Administration Assistant	Executive	JenniferWhalen	09/17/1995
Sales Representative	Sales	JonathonTaylor	03/24/2006
Sales Manager	Sales	JonathonTaylor	01/01/2007
Public Accountant	Executive	JenniferWhalen	07/01/2002

- Setelah itu, buat table baru untuk menyalin data yang telah di tampilkan sebelumnya. Gunakan subquery **INSERT** untuk melakukan penyalinan.

Untuk memudahkan, query pembuatan table baru ada di bawah ini.

```
CREATE TABLE kegiatan_1
( job_title          VARCHAR2(50)
, department_name    VARCHAR2(50)
, name               VARCHAR2(50)
, hire_date          DATE
) ;
```

- Selanjutnya, ubahlah hire_date pada table kegiatan_1 yang mulai bekerja pada tahun 1995 (dari 1 januari 1995 hingga 31 desember 1995) menjadi dimulai bekerja mulai tahun 2003 dengan statement subquery **UPDATE** kemudian tunjukkan.
- Terakhir, hapus record data pada table kegiatan_1 yang hire_date pada tahun 2003 (dari 1 januari 2003 hingga 31 desember 2003) dengan statement subquery **DELETE** kemudian tunjukkan.

KEGIATAN 2

Buatlah view untuk menampilkan nama, email, nomer telepon , dan gaji karyawan yang mulai bekerja pada tahun 2000.

RUBRIK PENILAIAN

Kegiatan 1 : 60%

- Select : 15%
- Insert : 10%
- Update : 10%
- Delete : 10%

Kegiatan 2 : 40%