

# SQL Lanjut

TPI4210  
Sistem dan Teknologi Informasi

# The GROUP BY Clause

- Syntax:

GROUP BY <nama kolom>

- Contoh

**ORDER\_LINE(ORDER\_NR, PRODUCT\_NR, QTY\_ON\_ORDER)**

```
SELECT PRODUCT_NR,  
SUM(QTY_ON_ORDER)  
FROM ORDER_LINE  
GROUP BY PRODUCT_NR;
```

Menampilkan jumlah tiap produk yang sedang dalam pesanan

# The HAVING Clause

- HAVING adalah SELECT untuk grup.
- HAVING memberi fasilitas untuk menerapkan persyaratan dalam grup yang terlibat dalam operasi

```
SELECT PRODUCT_NR,  
SUM(QTY_ON_ORDER)  
FROM ORDER_LINE  
GROUP BY PRODUCT_NR  
HAVING COUNT(*) < 20;
```

Akan menampilkan jumlah tiap produk dalam pesanan yang tingkat penjualannya tidak terlalu baik (dalam contoh ini produk terjual kurang dari 20 buah)

# Again, our sample base table: pupil

Surname	Initials	Suburb	Year_mark
Green	GG	Oakwood	56
Brown	BB	Elmwood	87
Gold	GG	Ashwood	68
White	WW	Elmwood	59
Violet	VV	Ashwood	77
Blue	BB	Elmwood	76
Black	BB	Oakwood	80
Red	RR	Oakwood	89
Orange	OO	Ashwood	75
Yellow	YY	Oakwood	64

# GROUP BY Example 1

```
select suburb,count(*) as  
number_of_pupils  
from pupil  
group by suburb;
```

Suburb	Number_of_pup
Ashwood	3
Elmwood	3
Oakwood	4

User:

Dari suburb mana saja siswa berasal, dan berapa jumlahnya

# GROUP BY Example 2

```
select suburb,count(*) as  
number_of_pupils from pupil  
group by suburb  
having count(*) < 4;
```

Suburb	Number_of_pup
Ashwood	3
Elmwood	3

User:

Dari suburb mana kita mendapat sedikit siswa (dalam contoh ini kurang dari 4)

# GROUP BY Example 3

```
select suburb,count(*) as  
number_of_pupils,  
avg(year_mark) as average from pupil  
group by suburb;
```

Suburb	Number_of_pup	Average
Ashwood	3	73,333
Elmwood	3	74,000
Oakwood	4	72,250

User:

Dari suburb dimana siswa berasal, berapa nilai rata-rata seluruh siswa di suburb tersebut

# The UPDATE Command

Bentuk umum:

```
UPDATE <table name>  
SET <column name> = <value>  
WHERE <selection condition>
```



# UPDATE Example

```
update pupil
set year_mark = 86
where surname = 'Green'
and suburb = 'OAKWOOD'
```

```
select *
from pupil
```

Surname	Initials	Suburb	Year_mark
Green	GG	Oakwood	86
Brown	BB	Elmwood	87
Gold	GG	Ashwood	68
White	WW	Elmwood	59
Violet	VV	Ashwood	77
Blue	BB	Elmwood	76
Black	BB	Oakwood	80
Red	RR	Oakwood	89
Orange	OO	Ashwood	75
Yellow	YY	Oakwood	64

# ALTER TABLE

- ALTER dapat digunakan untuk menambah kolom atau memodifikasi kolom yang sudah ada
- ALTER juga dapat digunakan untuk modifikasi integrity constraint

# ALTER TABLE

Aturan penambahan kolom:

- Kolom dapat ditambahkan kapan saja jika NOT NULL tidak didefinisikan
- Kolom dengan atribut NOT NULL tidak dapat ditambahkan dalam tabel yang tidak kosong (bisa menambah kolom tanpa mendefinisikan NOT NULL, dan kemudian dimodifikasi dengan NOT NULL)

Aturan modifikasi kolom:

- Lebar kolom character dapat ditambah kapan saja
- Jumlah digit dalam kolom NUMBER dapat ditambah kapan saja
- Jumlah angka decimal dalam kolom NUMBER dapat ditambah dan dikurangi kapan saja
- Tipe data suatu kolom dapat diubah hanya jika seluruh entrinya NULL
- Lebar kolom character dapat dikurangi hanya jika seluruh entrinya NULL
- Jumlah digit dalam kolom NUMBER dapat dikurangi jika seluruh entrinya NULL

# ALTER TABLE examples

- alter table pupil modify  
(given\_name varchar2(40));
- alter table pupil modify  
(surname NULL);
- alter table pupil  
add (birth\_date date, student\_id number);
- alter table customer  
add primary key (customer\_id);

# The CREATE TABLE Statement

2 bentuk perintah CREATE TABLE .....

```
create table tablename  
(columnname datatype  
column_constraint  
{,columnname data type  
column_constraint})  
[table_constraint]
```

```
create table tablename  
[(columnname {,columnname})]  
as subselect  
[with-clause]
```

Jika as digunakan, tabel baru yang dibuat mengambil nama kolom dan format dari tabel dalam subselect

# CREATE TABLE Example

```
create table student  
(stunum varchar2(3),  
surname varchar2(10),  
street varchar2(10),  
primary key (stunum));
```

```
create table employee  
(emp_id char(5) primary key,  
surname varchar2(30) not null,  
initials varchar2(4) not null,  
age number check (age between 18 and 65)  
dept_id number references  
department(dept_id));
```

# CREATE TABLE Example

```
create table employee  
(emp_id char(5),  
surname varchar2(30) not null,  
initials varchar2(4) not null,  
age number check (age between 18 and 65)  
dept_id number references  
department(dept_id),  
constraint employee_pk primary key  
(emp_id));
```

# CREATE TABLE Example

```
create table order_line  
(ord_id number constraint fk1  
references  
order(ord_id),  
prod_id number constraint fk2  
references  
product(prod_id),  
quantity number,  
constraint ol_pk primary key  
(ord_id,prod_id));
```



# CREATE TABLE Example

```
create table booking  
( tour_id number,  
  cust_id number,  
  dep_date date,  
  constraint booking_pk primary key  
  (tour_id,cust_id,dep_date),  
  constraint booking_fk1 foreign key  
  (tour_id,dep_date) references departure  
  (tour_id, dep_date) on delete cascade,  
  constraint booking_fk2 foreign key  
  (cust_id)  
  references customer (cust_id));
```

# CREATE TABLE Example

```
select *  
from student
```

stunum	surname	street
S23	Brown	Elm
S56	Violet	Oak
S76	Green	Ash
S45	Blue	Oak
S18	Azure	Ash
S78	Gold	Oak
S92	Black	Ash
S33	Red	Elm
S46	Orange	Elm
S21	Emerald	Oak

# CREATE TABLE Example

```
create table student_2  
as (select *  
from student)
```

Membuat tabel yang  
sama dengan tabel  
dalam subselect

```
select *  
from student_2
```

stunum	surname	street
S23	Brown	Elm
S56	Violet	Oak
S76	Green	Ash
S45	Blue	Oak
S18	Azure	Ash
S78	Gold	Oak
S92	Black	Ash
S33	Red	Elm
S46	Orange	Elm
S21	Emerald	Oak

# CREATE TABLE Example

```
create table student_3  
as (select surname, stunum  
from student)
```

Membuat tabel yang merupakan subset vertikal dari tabel dalam subselect

```
select *  
from student_3
```

surname	stunum
Brown	823
Violet	856
Green	876
Blue	845
Azure	818
Gold	878
Black	892
Red	833
Orange	846
Emerald	821

# Working with Views - 1

## CREATE VIEW Syntax

Syntax:

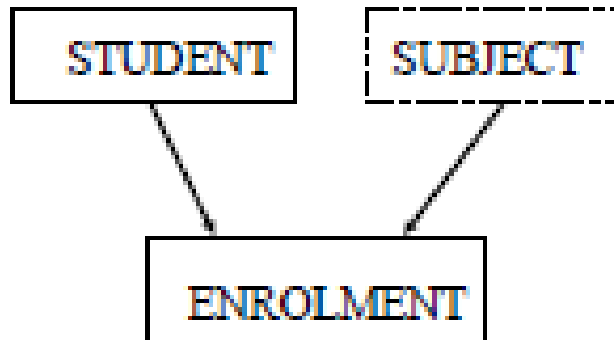
```
create view view_name [(columnname {,columnname})]  
as subselect
```

- Tujuan:  
Membuat tabel virtual untuk tampilan user tertentu dalam database.  
Dengan pilihan columnname, view dapat dibuat berbeda dari tabel asal
- Saat membuat view, data tidak diambil dari tabel asal. View didefinisikan dalam catalog dan digunakan untuk menampilkan data dari tabel asal saat nama view digunakan dalam operasi
- Tidak semua view dapat di update, sehingga meskipun data dapat diambil dan diupdate, tetapi hanya berlaku pada view, tidak pada tabel asal

# Working with Views - 2

## Setting the Scene

Data model:



```
create table enrolment  
(stunum varchar2(3),  
subject varchar2(7),  
result varchar2(2))
```

Perhatikan dua foreign key,  
stunum dan subject

# Working with Views - 3

## The CREATE VIEW Statement

```
create view student_data  
(student_id, last_name, subject, grade)  
as(select s.stunum, surname, subject,result  
from student s, enrolment e  
where s.stunum = e.stunum)
```

Dengan menggunakan daftar columnname, dapat dibuat sebuah view yang jauh lebih informatif bagi user

The view, (a virtual table), can now be retrieved just like a base table . . .

**select \***  
**from enrolment**

stunum	subject	result
S45	COT7710	HD
S92	COT7700	F
S23	COT7710	D
S18	COT7710	HD
S18	COT7700	F
S92	COT7710	D

**select \***  
**from student\_data**

student	last_name	subject	grade
S18	Azure	COT7700	F
S18	Azure	COT7710	HD
S23	Brown	COT7710	D
S45	Blue	COT7710	HD
S92	Black	COT7700	F
S92	Black	COT7710	D