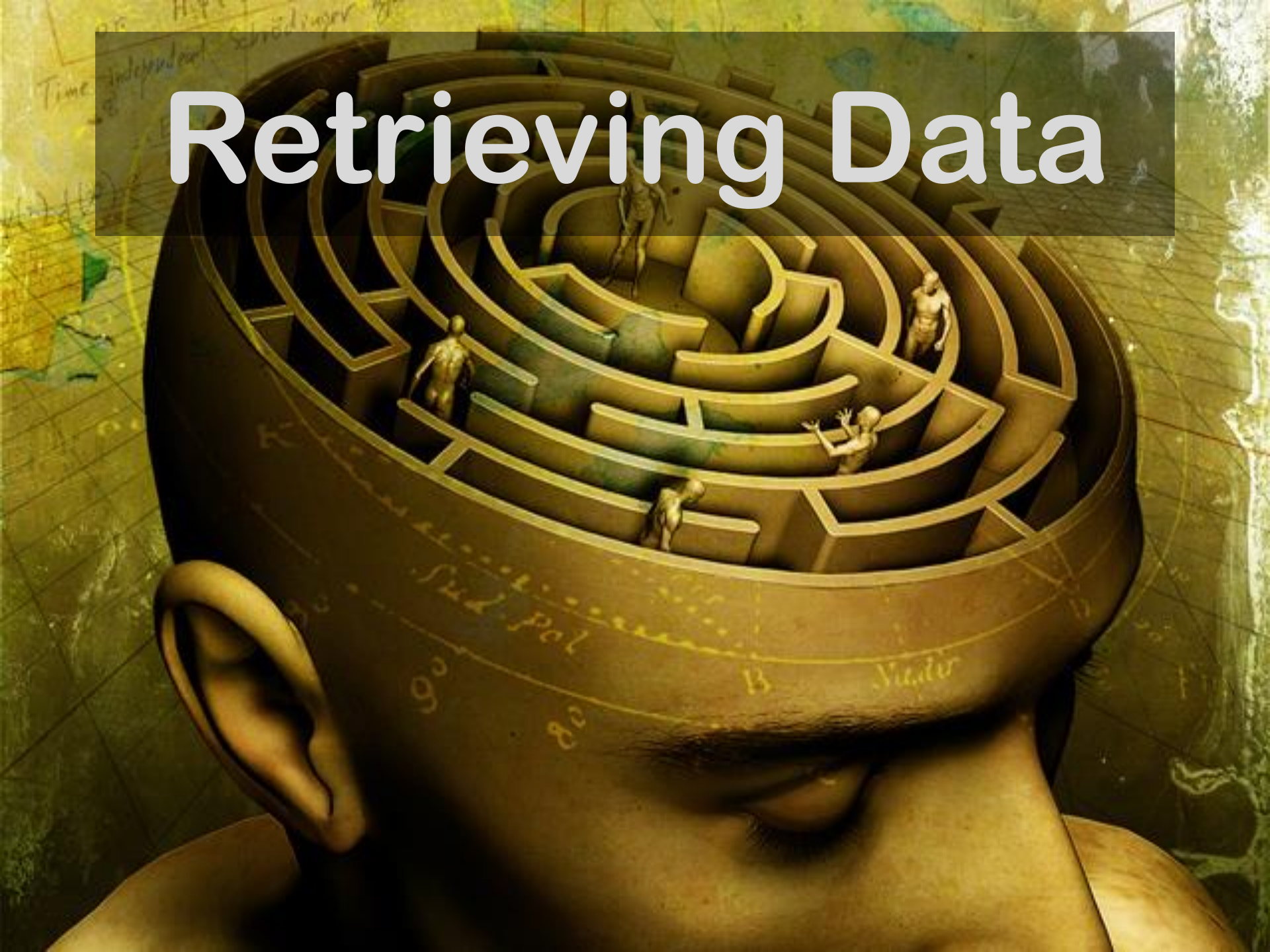
Retrieving Data



SQL Statements

SELECT statement mengambil (retrieves) informasi dari database

SELECT Statements

Projection

- Kemampuan dapat memilih kolom yang ingin ditampilkan dengan *query*

Selection

- Kemampuan untuk memilih baris yang ingin ditampilkan dengan *query*

Joining

- Kemampuan untuk menampilkan data yang tersimpan di tabel yang berbeda dengan menciptakan keterkaitan diantaranya

Aljabar Relational Review

Projection

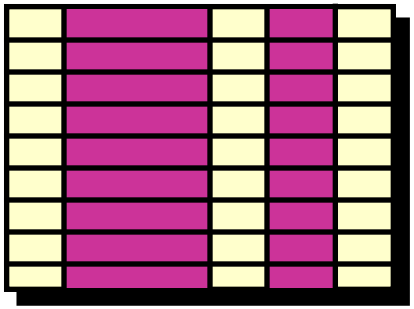


Table 1

Selection

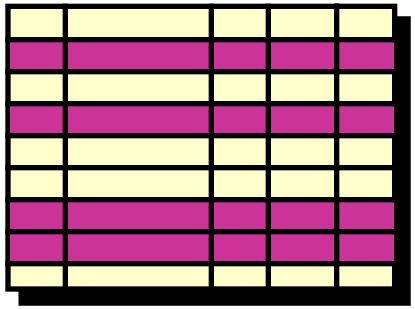


Table 1

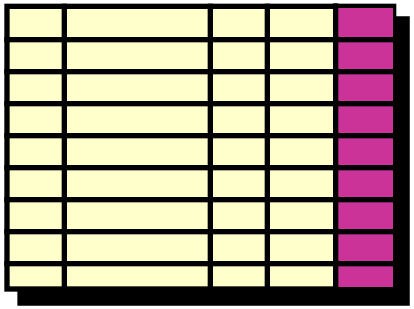


Table 1

Join

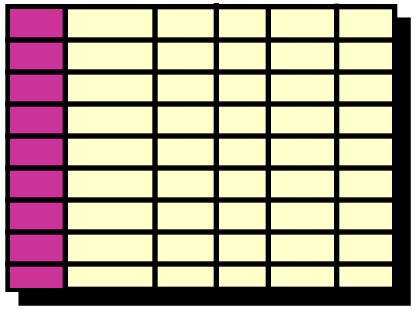


Table 2

Basic SELECT Statement

```
SELECT   *|{[DISTINCT] column|expression [alias],...}
FROM     table;
```

- SELECT mengidentifikasi columns
- FROM mengidentifikasi table

Select

REVIEW

Menampilkan data dengan kondisi data tertentu dengan klausa WHERE

```
SELECT * FROM namatabel WHERE kondisi;
```

Autocomplete: [Tab]->Next Tag. [Ctrl+Space]->List Matchin

```
1   SELECT * FROM jenisfilm WHERE jenis='Action';
```

1 Result | 2 Profiler | 3 Messages | 4 Ta

(Read Only)

kode	jenis	harga
F001	Action	90000
F002	Action	70000
F004	Action	800000

Beberapa operator perbandingan yang dapat digunakan pada klausa WHERE

= (sama dengan)

> (lebih dari)

< (kurang dari)

< > (tidak sama dengan)

>= (lebih dari atau sama dengan)

<= (kurang dari atau sama dengan).

Adapun operator lain, yaitu : AND, OR, NOT, BETWEEN-AND, IN dan LIKE

Tabel Jenis Film

	kode	jenis	harga
	F001	Action	25000
	F002	Horor	10000
	F003	Drama	40000
	F004	Komedi	30000
*	(NULL)	komedi	0

Operator AND

Menampilkan data pada tabel jenisfilm dimana nilai harga **berkisar dari 10000 hingga 30000**

Query | Query Builder | Schema Designer

Autocomplete: [Tab]->Next Tag. [Ctrl+Space]->List Matching Tags. [Ctrl+Enter]->Lis

```
1   SELECT * FROM jenisfilm WHERE harga>=10000 AND harga<=30000
```

1 Result | 2 Profiler | 3 Messages | 4 Table Data | 5 Info

(Read Only)

	kode	jenis	harga
	F001	Action	25000
	F002	Horor	10000
	F004	Komedi	30000

Operator OR

Menampilkan data pada tabel jenisfilm dimana nilai harga **sama dengan 10000 atau 30000**

Autocomplete: [Tab]->Next Tag. [Ctrl+Space]->List Matching Tags. [Ctrl+

```
1   SELECT * FROM jenisfilm WHERE harga=10000 OR harga=30000
```

1 Result | 2 Profiler | 3 Messages | 4 Table Data

(Read Only)

kode	jenis	harga
F002	Horor	10000
F004	Komedi	30000

Operator NOT

Menampilkan data pada tabel jenisfilm dimana nilai pada kolom jenis **tidak sama dengan action**

Autocomplete: [Tab]->Next Tag. [Ctrl+Space]->List Matching T

```
1   SELECT * FROM jenisfilm WHERE NOT jenis='Action'
```

1 Result | 2 Profiler | 3 Messages | 4 Table

(Read Only)

kode	jenis	harga
F002	Horor	10000
F003	Drama	40000
F004	Komedi	30000

Autocomplete: [Tab]->Next Tag. [Ctrl+Space]->List Matchin

```
1   SELECT * FROM jenisfilm WHERE jenis<>'Action'
```

1 Result | 2 Profiler | 3 Result | 4 Messa

(Read Only)

kode	jenis	harga
F002	Horor	10000
F003	Drama	40000
F004	Komedi	30000

Tabel Jenis Film

	kode	jenis	judul	harga
	F001	Action	Spiderman	25000
	F002	Horor	Suster Ngesot	10000
	F003	Drama	Love Story	40000
	F004	Komedi	21 Jump Street	30000

Operator LIKE

Menampilkan data pada tabel film dimana data pada kolom tertentu diawali dengan nilai tertentu, misalnya pada kolom judul dimana diawali dengan karakter ‘S’

Autocomplete: [Tab]->Next Tag. [Ctrl+Space]->List Matchin

```
1   SELECT * FROM jenisfilm WHERE judul LIKE 's%'
```

1 Result | 2 Profiler | 3 Messages | 4 Ta

(Read Only)

kode	jenis	judul	harga
F001	Action	Spiderman	25000
F002	Horor	Suster Ngesot	10000

Tabel Jenis Film

jenis	harga
Action	25000
Horor	10000
Drama	40000
Komedi	30000

Tabel Film

kode_film	jenis	judul	jumlahkeping	jumlahfilm
A01	Action	Spiderman	2	3
D01	Drama	Love Story	2	3
H01	Horor	Suster Ngesot	2	3

Menampilkan Data Lebih dari Dua Tabel

SELECT * from namatabel1,namatabel2,namatabel-n;

Autocomplete: [Tab]->Next Tag. [Ctrl+Space]->List Matching Tags. [Ctrl+Enter]->List All Tags.

```
1   SELECT * FROM jenisfilm,film;
```

1 Result | 2 Profiler | 3 Messages | 4 Table Data | 5 Info | 6 History

(Read Only)

jenis	harga	kode_film	jenis	judul	jumlahkeping	jumlahfilm
Action	25000	A01	Action	Spiderman	2	3
Action	25000	D01	Drama	Love Story	2	3
Action	25000	H01	Horor	Suster Ngesot	2	3
Horor	10000	A01	Action	Spiderman	2	3
Horor	10000	D01	Drama	Love Story	2	3
Horor	10000	H01	Horor	Suster Ngesot	2	3
Drama	40000	A01	Action	Spiderman	2	3
Drama	40000	D01	Drama	Love Story	2	3
Drama	40000	H01	Horor	Suster Ngesot	2	3
Komedi	30000	A01	Action	Spiderman	2	3
Komedi	30000	D01	Drama	Love Story	2	3
Komedi	30000	H01	Horor	Suster Ngesot	2	3

Nested Queries/Subquery

Subquery berarti *query* di dalam *query*. Dengan menggunakan *subquery*, hasil dari *query* akan menjadi bagian dari *query* di atasnya.

Subquery terletak di dalam klausa **WHERE** atau **HAVING**.

Pada klausa **WHERE**, *subquery* digunakan untuk memilih baris-baris tertentu yang kemudian digunakan oleh *query*.

Sedangkan pada klausa **HAVING**, *subquery* digunakan untuk memilih kelompok baris yang kemudian digunakan oleh *query*.

Operator IN

Menampilkan data pada tabel jenisfilm yang mana
data pada kolom jenis-nya tercantum pada tabel film

Autocomplete: [Tab]->Next Tag. [Ctrl+Space]->List Matching Tags. [Ctrl+Enter]->

```
1  SELECT * FROM jenisfilm WHERE jenis IN (SELECT jenis FROM film)
```

1 Result | 2 Profiler | 3 Messages | 4 Table Data | 5 Info

(Read Only)

	jenis	harga
	Action	25000
	Horor	10000
	Drama	40000

Query

SubQuery

Operator NOT IN

Menampilkan data pada tabel jenisfilm yang mana data pada kolom jenis-nya tidak tercantum pada tabel film

Autocomplete: [Tab]->Next Tag. [Ctrl+Space]->List Matching Tags. [Ctrl+Enter]->List

```
1   SELECT * FROM jenisfilm WHERE jenis NOT IN (SELECT jenis FROM film)
```

1 Result | 2 Profiler | 3 Messages | 4 Table Data | 5 Info

(Read Only)

jenis	harga
Komedi	30000

Operator NOT EXIST

Menampilkan data pada tabel jenisfilm yang mana data pada kolom jenis-nya tidak tercantum pada tabel film

Autocomplete: [Tab]->Next Tag. [Ctrl+Space]->List Matching Tags. [Ctr

```
1   SELECT * FROM jenisfilm WHERE NOT EXISTS
2   (SELECT * FROM film WHERE jenisfilm.jenis = film.jenis)
```

1 Result | 2 Profiler | 3 Messages | 4 Table Data

(Read Only)

jenis	harga
Komedi	30000

Operator ANY

Operator ANY digunakan berkaitan dengan *subquery*. Operator ANY menghasilkan TRUE (benar) jika paling tidak salah satu perbandingan dengan hasil *subquery* menghasilkan nilai TRUE

Menampilkan semua data jenisfilm yang harganya bukan yang terkecil

Autocomplete: [Tab]->Next Tag. [Ctrl+Space]->List M

```
1   SELECT * FROM jenisfilm WHERE
2   harga> ANY (SELECT harga FROM jenisfilm)
```

1 Result | 2 Profiler | 3 Messages

(Read Only)

jenis	harga
Action	25000
Drama	40000
Komedi	30000

Operator ALL

Operator ALL digunakan untuk melakukan perbandingan dengan *subquery*. Kondisi dengan ALL menghasilkan nilai TRUE (benar) jika *subquery* tidak menghasilkan apapun atau jika perbandingan menghasilkan TRUE untuk setiap nilai *query* terhadap hasil *subquery*

Autocomplete: [Tab]->Next Tag. [Ctrl+Space]->List Ma

```
1   SELECT * FROM jenisfilm WHERE
2   harga>= ALL (SELECT harga FROM jenisfilm)
```

1 Result | 2 Profiler | 3 Messages

(Read Only)

jenis	harga
Drama	40000

Menampilkan data jenisfilm yang harganya paling tinggi

Tabel Film

kode_film	jenis	judul	jumlahkeping	jumlahfilm
A01	Action	Spiderman	2	3
A02	Action	Spiderman 2	2	5
A03	Action	Spiderman 3	2	4
D01	Drama	Love Story	2	3
H01	Horor	Suster Ngesot	2	3
H02	Horor	Evil Death	2	2

Klausa GROUP BY

Klausa GROUP BY digunakan untuk melakukan pengelompokan data

Menampilkan kolom jenis dan digabungkan dengan SUM(jml_film) yang dikelompokkan berdasarkan kolom jenis pada tabel film

Autocomplete: [Tab]->Next Tag. [Ctrl+Space]->List Matching Tags. [C

```
1   SELECT jenis,SUM(jumlahfilm) FROM film GROUP BY jenis
```

1 Result | 2 Profiler | 3 Messages | 4 Table Data

(Read Only)

jenis	sum(jumlahfilm)
Action	12
Drama	3
Horor	5

Klausa HAVING

Klausa HAVING digunakan untuk menentukan kondisi bagi klausa GROUP BY. Kelompok yang memenuhi HAVING saja yang akan dihasilkan.

Menampilkan data hanya kolom jenis yang dikelompokkan berdasarkan kolom jenis, dimana jumlah film berdasarkan kelompoknya harus lebih besar dari satu pada tabel film

Autocomplete: [Tab]->Next Tag. [Ctrl+Space]->List Matching Tags. [Ctrl+Enter]->

```
1   SELECT jenis FROM film GROUP BY jenis HAVING COUNT(kode_film)>1;
```

1 Result | 2 Profiler | 3 Messages | 4 Table Data | 5 Info

(Read Only)

jenis
Action
Horor