



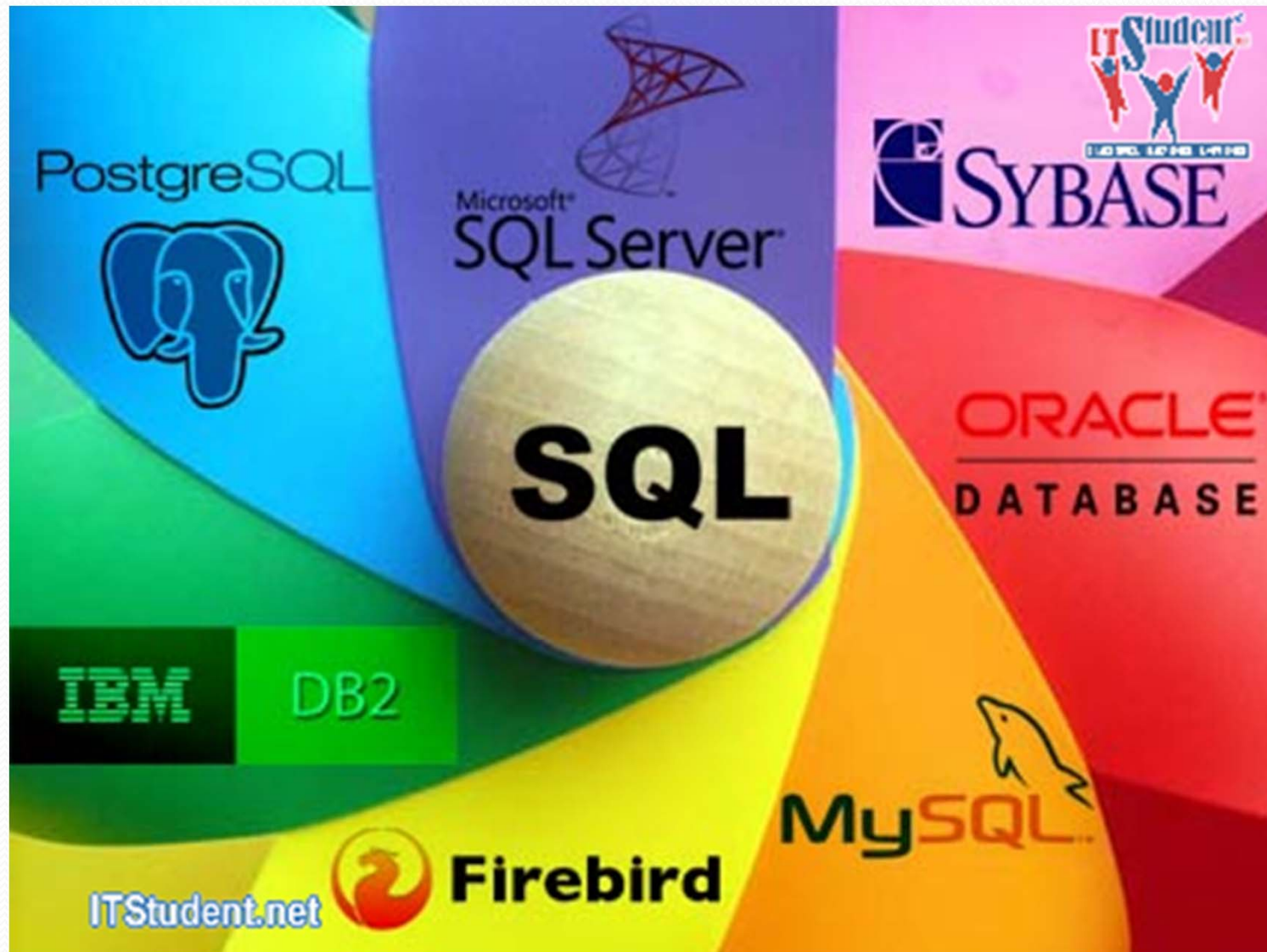
Pengaksesan Basis Data

SQL [1]

Fak. Teknik Prodi Teknik Informatika
Universitas Pasundan

Caca E. Supriana, S.Si., MT.

caca.e.supriana@unpas.ac.id
caca-e-supriana.blogspot.com





SQL

- IBM mengembangkan versi asli SQL di San Jose Research Laboratory (sekarang Almaden Research Center)
- Diberi nama SEQUEL, pada tahun 1970an sebagai bagian dari System R Project
- Sejak saat itu, bahasa sequel berevolusi dan berganti nama menjadi **Structure Query Language (SQL)**
- Di-support banyak aplikasi produk, SQL menjadi bahasa standard 'de facto' basis data relasional



Sejarah SQL

- Tahun 1986, ANSI (*American National Standards Institute*) dan ISO (*International Standard Organization*) mengumumkan standard SQL, SQL-86
- IBM merilis *Systems Application Architecture* SAA-SQL tahun 1987
- Berturut-turut ANSI merilis SQL-89, SQL- 92, dan SQL- 99
- ANSI terakhir merilis SQL-2003, digunakan pula oleh ISO yang beranggotakan 150 negara.

Standar SQL

1. Menentukan sintaks dan semantik dari definisi data SQL dan manipulasi bahasa.
2. Menentukan struktur data dan operasi dasar untuk merancang, mengakses, mempertahankan, mengendalikan, dan melindungi database SQL.
3. Menyediakan kakas untuk portabilitas dari definisi basis data dan penerapan modul antara sesuai DBMS.
4. Menentukan standar level minimal (Level 1) dan level lengkap (Level 2), yang memungkinkan derajat yang berbeda adopsi dalam produk DBMS.
5. Memberikan standar awal, meski tidak lengkap, yang akan ditingkatkan untuk menyertakan spesifikasi yang menangani topik-topik seperti integritas referensial, manajemen transaksi, user-defined, operator gabungan luar equi-join, dan set karakter nasional.



Keuntungan SQL [1]

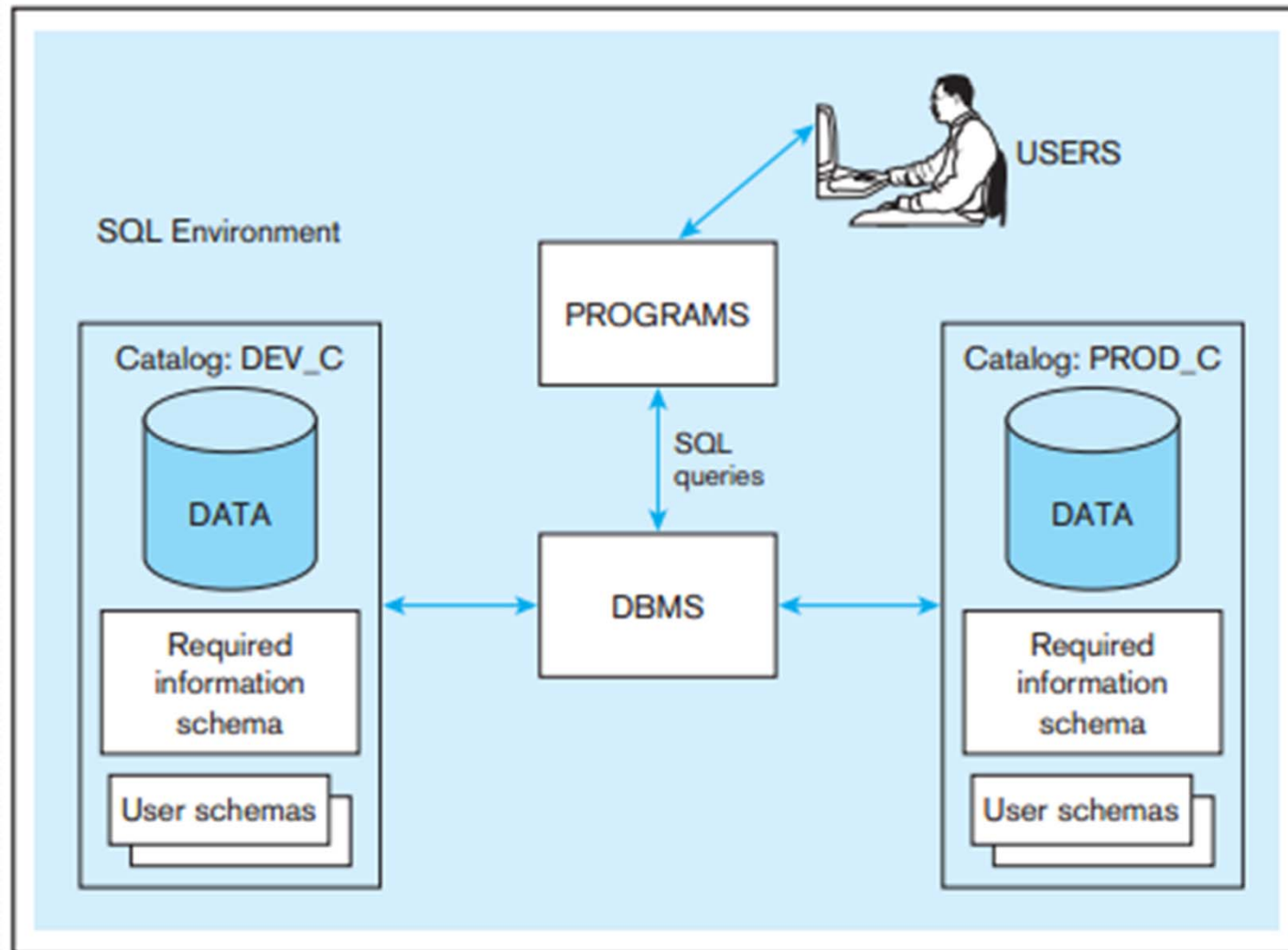
- **Biaya Pelatihan.** Mengurangi Pelatihan dalam suatu organisasi untuk berkonsentrasi pada satu bahasa. Para profesional bidang sistem informasi terlatih dalam bahasa yang sama mengurangi pelatihan kembali untuk karyawan baru.
- **Produktivitas.** Profesional bidang sistem informasi dapat belajar SQL secara menyeluruh dan menjadi mahir. Sebuah organisasi mampu untuk berinvestasi di alat untuk membantu IS profesional menjadi lebih produktif.
- **Portabilitas Aplikasi.** Portabilitas aplikasi dapat dipindahkan dari DBMS ke DBMS lain ketika setiap DBMS menggunakan SQL. Lebih ekonomis untuk industri perangkat lunak komputer untuk mengembangkan off-the-shelf software aplikasi ketika ada standar bahasa.



Keuntungan SQL [2]

- **Aplikasi Berumur Lebih Panjang.** Sebuah bahasa standar cenderung tetap sehingga untuk waktu yang lama; maka akan ada sedikit tekanan untuk menulis ulang aplikasi lama.
- **Mengurangi Ketergantungan Pada Satu Vendor.** Ketika bahasa hak milik digunakan, lebih mudah untuk menggunakan vendor yang berbeda untuk DBMS, pelatihan dan pelayanan pendidikan, aplikasi perangkat lunak, dan bantuan konsultasi; pasar untuk vendor tersebut akan lebih kompetitif, yang dapat menurunkan harga dan meningkatkan pelayanan.
- **Komunikasi Lintas Sistem.** DBMS yang berbeda dan aplikasi program dapat lebih mudah berkomunikasi dan bekerja sama dalam mengelola data dan program pengolah milik pengguna.

Lingkungan SQL



Fungsi SQL

- ***Data Definition Language*** (DDL)
 - Membuat objek database, seperti tabel, indeks, dan view
 - Menentukan hak akses terhadap objek database
- ***Data Manipulation Language*** (DML)
 - Termasuk perintah untuk memasukkan, update, menghapus, dan mengambil data dalam tabel database
- ***Data Command Language*** (DCL)
 - Perintah yang membantu DBA (*DB Administrator*) mengontrol database; termasuk perintah untuk memberikan atau mencabut hak istimewa untuk mengakses database atau objek tertentu dalam database dan untuk menyimpan atau menghapus transaksi yang akan mempengaruhi database.

SQL DDL Commands

TABLE
7.1

SQL Data Definition Commands

COMMAND OR OPTION	DESCRIPTION
CREATE SCHEMA AUTHORIZATION	Creates a database schema
CREATE TABLE	Creates a new table in the user's database schema
NOT NULL	Ensures that a column will not have null values
UNIQUE	Ensures that a column will not have duplicate values
PRIMARY KEY	Defines a primary key for a table
FOREIGN KEY	Defines a foreign key for a table
DEFAULT	Defines a default value for a column (when no value is given)
CHECK	Constraint used to validate data in an attribute
CREATE INDEX	Creates an index for a table
CREATE VIEW	Creates a dynamic subset of rows/columns from one or more tables
ALTER TABLE	Modifies a table's definition (adds, modifies, or deletes attributes or constraints)
CREATE TABLE AS	Creates a new table based on a query in the user's database schema
DROP TABLE	Permanently deletes a table (and thus its data)
DROP INDEX	Permanently deletes an index
DROP VIEW	Permanently deletes a view

SQL DML Commands [1]

TABLE 7.2 SQL Data Manipulation Commands

COMMAND OR OPTION	DESCRIPTION
INSERT	Inserts row(s) into a table
SELECT	Selects attributes from rows in one or more tables or views
WHERE	Restricts the selection of rows based on a conditional expression
GROUP BY	Groups the selected rows based on one or more attributes
HAVING	Restricts the selection of grouped rows based on a condition
ORDER BY	Orders the selected rows based on one or more attributes
UPDATE	Modifies an attribute's values in one or more table's rows
DELETE	Deletes one or more rows from a table
COMMIT	Permanently saves data changes
ROLLBACK	Restores data to their original values

SQL DML Commands [2]

TABLE
7.2

SQL Data Manipulation Commands (continued)

COMMAND OR OPTION	DESCRIPTION
COMPARISON OPERATORS	
=, <, >, <=, >=, <>	Used in conditional expressions
LOGICAL OPERATORS	
AND/OR/NOT	Used in conditional expressions
SPECIAL OPERATORS	Used in conditional expressions
BETWEEN	Checks whether an attribute value is within a range
IS NULL	Checks whether an attribute value is null
LIKE	Checks whether an attribute value matches a given string pattern
IN	Checks whether an attribute value matches any value within a value list
EXISTS	Checks whether a subquery returns any rows
DISTINCT	Limits values to unique values
AGGREGATE FUNCTIONS	Used with SELECT to return mathematical summaries on columns
COUNT	Returns the number of rows with non-null values for a given column
MIN	Returns the minimum attribute value found in a given column
MAX	Returns the maximum attribute value found in a given column
SUM	Returns the sum of all values for a given column
AVG	Returns the average of all values for a given column



Membuat Database

- 2 tugas harus diselesaikan dalam membuat database
 - Buat struktur database
 - Buat tabel yang akan menyimpan data pengguna akhir
- Tugas pertama:
 - RDBMS (Relational DBMS) membuat file fisik yang akan menyimpan database
 - Cenderung berbeda secara substansial dari satu RDBMS ke yang lain

Skema Database

- Otorisasi
 - Proses dimana DBMS memverifikasi bahwa hanya pengguna terdaftar dapat mengakses database
 - Log on ke RDBMS menggunakan user ID dan password yang dibuat oleh administrator database
- Skema
 - Kelompok obyek-seperti database sebagai tabel dan indeks-yang berhubungan satu sama lain



Data Types

- Jenis data seleksi biasanya ditentukan oleh sifat data dan dengan tujuan penggunaannya
- Memperhatikan penggunaan diharapkan atribut untuk menyortir dan tujuan pengambilan data

Data Types

TABLE
7.4

Some Common SQL Data Types

DATA TYPE	FORMAT	COMMENTS
Numeric	NUMBER(L,D)	The declaration NUMBER(7,2) indicates numbers that will be stored with two decimal places and may be up to six digits long, including the sign and the decimal place. Examples: 12.32, -134.99.
	INTEGER	May be abbreviated as INT. Integers are (whole) counting numbers, so they cannot be used if you want to store numbers that require decimal places.
	SMALLINT	Like INTEGER, but limited to integer values up to six digits. If your integer values are relatively small, use SMALLINT instead of INT.
	DECIMAL(L,D)	Like the NUMBER specification, but the storage length is a <i>minimum</i> specification. That is, greater lengths are acceptable, but smaller ones are not. DECIMAL(9,2), DECIMAL(9), and DECIMAL are all acceptable.
Character	CHAR(L)	Fixed-length character data for up to 255 characters. If you store strings that are not as long as the CHAR parameter value, the remaining spaces are left unused. Therefore, if you specify CHAR(25), strings such as "Smith" and "Katzenjammer" are each stored as 25 characters. However, a U.S. area code is always three digits long, so CHAR(3) would be appropriate if you wanted to store such codes.
	VARCHAR(L) or VARCHAR2(L)	Variable-length character data. The designation VARCHAR2(25) will let you store characters up to 25 characters long. However, VARCHAR will not leave unused spaces. Oracle users may use VARCHAR2 as well as VARCHAR.
Date	DATE	Stores dates in the Julian date format.



Membuat Struktur Tabel

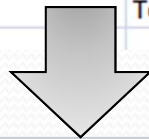
- Gunakan satu baris per kolom (atribut) definisi
- Gunakan spasi untuk berbaris karakteristik atribut dan kendala
- Tabel dan atribut nama dikapitalisasi
- NOT NULL spesifikasi
- spesifikasi UNIK



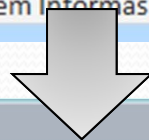
Membuat Struktur Tabel

- Atribut kunci primer mengandung *NOT NULL* dan spesifikasi UNIK
- RDBMS otomatis akan menegakkan integritas referensial untuk kunci asing (*FOREIGN KEY*)
- Urutan perintah diakhiri dengan titik koma (;)

Kuliah			
	Field Name	Data Type	
	kode_mk	Text	field size 15 [Primary Key]
	nama_mk	Text	field size 50
	sks	Number	field size Long Integer
	semester	Text	field size 20



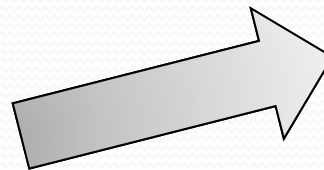
Kuliah				
	kode_mk	nama_mk	sks	semester
+	mk001	Sistem Basis Data 1	3	2
+	mk002	Sistem Operasi	3	2
+	mk003	Optimasi Basis Data	2	5
+	mk004	Sistem Informasi	3	4



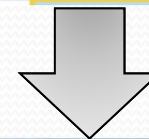
Kuliah

*
kode_mk
nama_mk
sks
semester

Field:	kode_mk	nama_mk
Table:	Kuliah	Kuliah
Sort:		
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Criteria:		



Query2	
kode_mk	nama_mk
mk001	Sistem Basis Data 1
mk002	Sistem Operasi
mk003	Optimasi Basis Data
mk004	Sistem Informasi



```
SELECT Kuliah.kode_mk, Kuliah.nama_mk
FROM Kuliah;
```



SQL Statements

- **SELECT** → memilih satu, beberapa atau semua field dari satu atau lebih tabel
- **FROM** → memilih tabel yang field nya dipilih dalam perintah select di atas
- **[WHERE]** → merupakan syarat dari field, tabel atau relasi antar tabel
- **[GROUP BY]** → mengelompokkan field mengacu pada field tertentu

Membuat Database

```
mysql> create database perkuliahan;
```

```
Query OK, 1 row affected (0.10 sec)
```

```
mysql> use perkuliahan;
```

```
Database changed
```

Membuat Tabel

```
mysql> create table kuliah(  
    -> kode_mk char(5) primary key not null,  
    -> nama_mk char(25) not null,  
    -> sks int not null,  
    -> semester char(2));
```

Query OK, 0 rows affected (0.36 sec)

```
mysql> desc kuliah;
```

Field	Type	Null	Key	Default	Extra
kode_mk	char(5)	NO	PRI		
nama_mk	char(25)	NO			
sks	int(11)	NO			
semester	char(2)	YES		NULL	

4 rows in set (0.01 sec)

Mengisi Tabel

```
mysql> insert into kuliah
-> (kode_mk, nama_mk, sks, semester)
-> values
-> ('IF01', 'SBD1', 3, 2),
-> ('IF02', 'SBD2', 3, 3),
-> ('IF03', 'Aplikasi Perkantoran', 2, 1),
-> ('IF04', 'Optimasi Basis Data', 2, 5),
-> ('IF05', 'Sistem Informasi', 3, 4);
```

Query OK, 5 rows affected (0.04 sec)

Records: 5 Duplicates: 0 Warnings: 0

Melihat Isi Tabel

```
mysql> select * from kuliah;
```

kode_mk	nama_mk	sks	semester
IF01	SBD1	3	2
IF02	SBD2	3	3
IF03	Aplikasi Perkantoran	2	1
IF04	Optimasi Basis Data	2	5
IF05	Sistem Informasi	3	4

```
5 rows in set (0.00 sec)
```


Contoh 1 SQL

DB_Penduduk							
no_ktp	nama	alamat	no_telpon	jenis_kelam	pekerjaan	pendidikan	
01222345323	Asep	Cisatu I/12 Bandung 401223	(022)11234234	laki-laki	pegawai swasta	sarjana	
02235647737	Badu	Cisatu II/45b Bandung 401223	(022)55466737	laki-laki	pegawai negeri	sma	
02236373662	Citra	Meleeng 23 Bandung 4012231	(022)66474663	perempuan	pegawai swasta	diploma	
03344774622	Cahyadi	Brantak 2 Bandung 4033562	(022)66739987	laki-laki	militer	sma	
07787677867	Doddy	Balubur 7 Bandung 4678875	(022)66437883	laki-laki	wiraswasta	sarjana	
09878867772	Diana	Meleeng 34a Bandung 401223:	(022)66437882	perempuan	pegawai swasta	sarjana	
*							

- Tampilkan seluruh *field* dari tabel DB_Penduduk

SELECT *

FROM DB_Penduduk;

{hasilnya sama dengan tabel diatas}

Contoh 2 SQL

DB_Penduduk							
no_ktp	nama	alamat	no_telpon	jenis_kelam	pekerjaan	pendidikan	
01222345323	Asep	Cisatu I/12 Bandung 401223	(022)11234234	laki-laki	pegawai swasta	sarjana	
02235647737	Badu	Cisatu II/45b Bandung 401223	(022)55466737	laki-laki	pegawai negeri	sma	
02236373662	Citra	Meleeng 23 Bandung 4012231	(022)66474663	perempuan	pegawai swasta	diploma	
03344774622	Cahyadi	Brantak 2 Bandung 4033562	(022)66739987	laki-laki	militer	sma	
07787677867	Doddy	Balubur 7 Bandung 4678875	(022)66437883	laki-laki	wiraswasta	sarjana	
09878867772	Diana	Meleeng 34a Bandung 401223	(022)66437882	perempuan	pegawai swasta	sarjana	
*							

- Tampilkan seluruh *field* dari tabel DB_Penduduk untuk penduduk bernama “Asep”

SELECT *

FROM DB_Penduduk

WHERE nama = “Asep”;

DB_Penduduk Query1							
no_ktp	nama	alamat	no_telpon	jenis_kelam	pekerjaan	pendidikan	
01222345323	Asep	Cisatu I/12 Bandung 401223	(022)11234234	laki-laki	pegawai swasta	sarjana	
*							

Contoh 3 SQL

DB_Penduduk							
no_ktp	nama	alamat	no_telpon	jenis_kelam	pekerjaan	pendidikan	
01222345323	Asep	Cisatu I/12 Bandung 401223	(022)11234234	laki-laki	pegawai swasta	sarjana	
02235647737	Badu	Cisatu II/45b Bandung 401223	(022)55466737	laki-laki	pegawai negeri	sma	
02236373662	Citra	Meleeng 23 Bandung 4012231	(022)66474663	perempuan	pegawai swasta	diploma	
03344774622	Cahyadi	Brantak 2 Bandung 4033562	(022)66739987	laki-laki	militer	sma	
07787677867	Doddy	Balubur 7 Bandung 4678875	(022)66437883	laki-laki	wiraswasta	sarjana	
09878867772	Diana	Meleeng 34a Bandung 401223	(022)66437882	perempuan	pegawai swasta	sarjana	
*							

- Tampilkan *field* no ktp & nama dari tabel DB_Penduduk untuk penduduk bernama “Asep”

```
SELECT no_ktp, nama  
FROM DB_Penduduk  
WHERE nama = “Asep”;
```

DB_Penduduk		Query1	
no_ktp	nama		
01222345323	Asep		
*			

Contoh 4 SQL

DB_Penduduk							
no_ktp	nama	alamat	no_telpon	jenis_kelamin	pekerjaan	pendidikan	
01222345323	Asep	Cisatu I/12 Bandung 401223	(022)11234234	laki-laki	pegawai swasta	sarjana	
02235647737	Badu	Cisatu II/45b Bandung 401223	(022)55466737	laki-laki	pegawai negeri	sma	
02236373662	Citra	Meleeng 23 Bandung 4012231	(022)66474663	perempuan	pegawai swasta	diploma	
03344774622	Cahyadi	Brantak 2 Bandung 4033562	(022)66739987	laki-laki	militer	sma	
07787677867	Doddy	Balubur 7 Bandung 4678875	(022)66437883	laki-laki	wiraswasta	sarjana	
09878867772	Diana	Meleeng 34a Bandung 401223	(022)66437882	perempuan	pegawai swasta	sarjana	
*							

- Tampilkan *field* no ktp & nama dari tabel DB_Penduduk untuk penduduk dengan jenis kelamin laki-laki dan pekerjaannya adalah pegawai swasta

```
SELECT no_ktp, nama
FROM DB_Penduduk
WHERE jenis_kelamin = "laki-laki"
AND pekerjaan = "pegawai negeri";
```

DB_Penduduk		Query1	
no_ktp	nama		
02235647737	Badu		
*			

Contoh 5 SQL

DB_Penduduk							
no_ktp	nama	alamat	no_telpon	jenis_kelam	pekerjaan	pendidikan	
01222345323	Asep	Cisatu I/12 Bandung 401223	(022)11234234	laki-laki	pegawai swasta	sarjana	
02235647737	Badu	Cisatu II/45b Bandung 401223	(022)55466737	laki-laki	pegawai negeri	sma	
02236373662	Citra	Meleeng 23 Bandung 4012231	(022)66474663	perempuan	pegawai swasta	diploma	
03344774622	Cahyadi	Brantak 2 Bandung 4033562	(022)66739987	laki-laki	militer	sma	
07787677867	Doddy	Balubur 7 Bandung 4678875	(022)66437883	laki-laki	wiraswasta	sarjana	
09878867772	Diana	Meleeng 34a Bandung 401223	(022)66437882	perempuan	pegawai swasta	sarjana	
*							

- Tampilkan seluruh *field* dari tabel DB_Penduduk untuk penduduk yang namanya dimulai huruf “D”

SELECT *

FROM DB_Penduduk

WHERE nama like “D*”;

DB_Penduduk Query1							
no_ktp	nama	alamat	no_telpon	jenis_kelam	pekerjaan	pendidikan	
07787677867	Doddy	Balubur 7 Bandung 4678875	(022)66437883	laki-laki	wiraswasta	sarjana	
09878867772	Diana	Meleeng 34a Bandung 401223	(022)66437882	perempuan	pegawai swasta	sarjana	

Latihan

1. Tampilkan nama dan alamat penduduk.
2. Tampilkan nama dan alamat penduduk bernama “Diana”.
3. Cari nomor ktp dan nama penduduk yang pekerjaannya pegawai negeri.
4. Cari nama dan alamat penduduk yang pekerjaannya pegawai negeri dan berpendidikan sarjana.
5. Cari nama, no telpon dan pekerjaan penduduk yang tinggal di Cisatu.

DB_Penduduk							
no_ktp	nama	alamat	no_telpon	jenis_kelam	pekerjaan	pendidikan	
01222345323	Asep	Cisatu I/12 Bandung 401223	(022)11234234	laki-laki	pegawai swasta	sarjana	
02235647737	Badu	Cisatu II/45b Bandung 401223	(022)55466737	laki-laki	pegawai negeri	sma	
02236373662	Citra	Meleeng 23 Bandung 4012231	(022)66474663	perempuan	pegawai swasta	diploma	
03344774622	Cahyadi	Brantak 2 Bandung 4033562	(022)66739987	laki-laki	militer	sma	
07787677867	Doddy	Balubur 7 Bandung 4678875	(022)66437883	laki-laki	wiraswasta	sarjana	
09878867772	Diana	Meleeng 34a Bandung 401223	(022)66437882	perempuan	pegawai swasta	sarjana	
*							