

Administrasi Basis Data

Yoannita



Data

- Kumpulan dari angka-angka maupun karakter-karakter yang tidak memiliki arti.
- Fakta mengenai objek, orang, dan lainnya.
- Data dinyatakan dengan nilai (angka, deretan karakter, atau simbol)
- Data → **bahan** dalam suatu proses pengolahan data
- Contoh data adalah catatan identitas pegawai, catatan transaksi pembelian, dll;
- Data dapat diolah sehingga menghasilkan **informasi**.

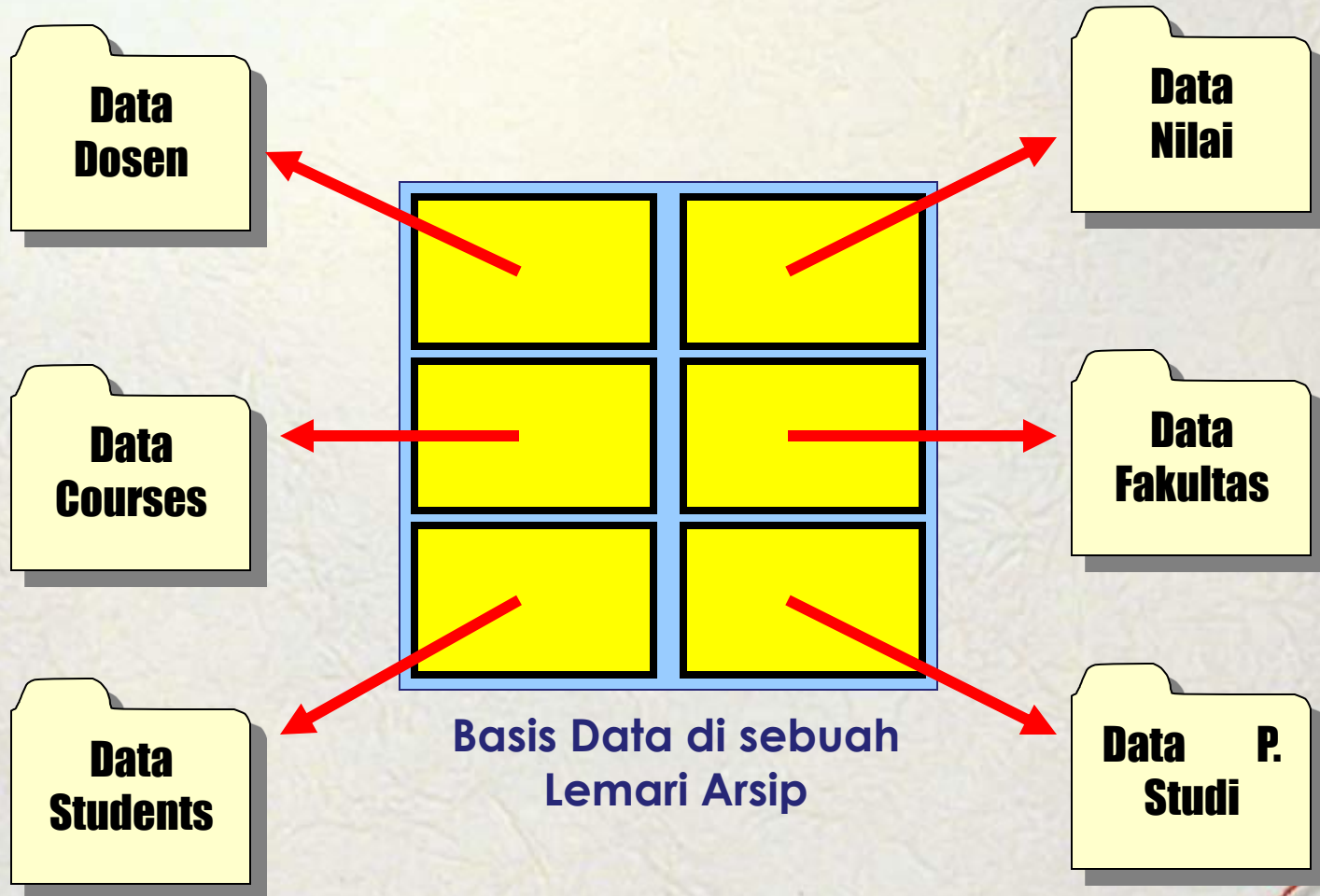
Informasi

- Hasil analisis dan sintesis terhadap data
- Data yang telah diorganisasikan ke dalam bentuk yang sesuai dengan kebutuhan seseorang
- Informasi → hasil pengolahan data sehingga menjadi bentuk yang penting bagi penerimanya dan mempunyai kegunaan sebagai dasar dalam pengambilan keputusan.
- Contoh informasi : daftar pegawai berdasarkan departemen, rekapitulasi transaksi pembelian pada akhir bulan.

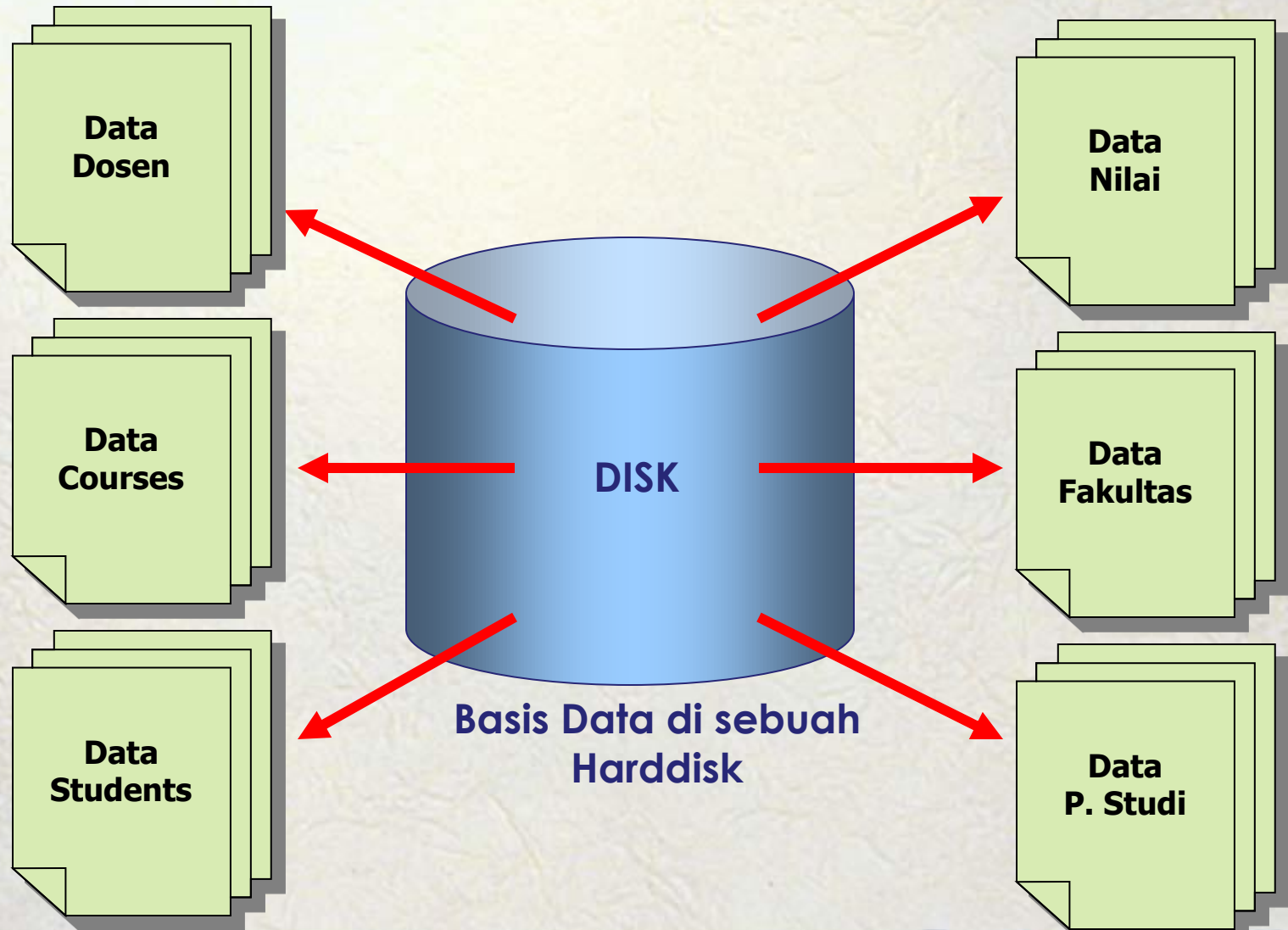
Basis Data

- Kumpulan file/tabel/arsip yang saling berhubungan yang disimpan dalam media penyimpanan elektronik.

Basis Data



Basis Data



Basis Data

- Himpunan kelompok data yang saling berhubungan yang diorganisasi sedemikian rupa sehingga kelak dapat dimanfaatkan dengan cepat dan mudah
- Kumpulan data yang saling berhubungan yang disimpan secara bersama tanpa pengulangan (redundancy) yang tidak perlu, untuk memenuhi kebutuhan.

Manfaat Basis Data

1. Kecepatan dan kemudahan (speed)
2. Kebersamaan pemakai (shareability)
3. Pemusatan kontrol data
4. Efisiensi ruang penyimpanan (space)
5. Keakuratan (accuracy)
6. Ketersediaan (availability)

Manfaat Basis Data

7. Kelengkapan (completeness)
8. Keamanan (security)
9. Kemudahan dalam pembuatan program aplikasi baru
10. Pemakaian secara langsung
11. Kebebasan data (Data independence)
12. User View

Operasi Basis Data

- Pembuatan basis data
- Penghapusan basis data
- Pembuatan file/table
- Penghapusan file/table
- Pengubahan table
- Penambahan/pengisian
- Pengambilan data
- Penghapusan data

Database Management System (DBMS)

- *Collection of interrelated data*
- *Set of programs to access the data*
- *DBMS contains information about a particular enterprise*
- *DBMS provides an environment that is both convenient and efficient to use.*

- **Sistem manajemen basis data** (*database management system, DBMS*): suatu sistem atau perangkat lunak yang dirancang untuk mengelola suatu basis data dan menjalankan operasi terhadap data yang diminta banyak pengguna.
- Contoh SMDB : **Oracle, SQL server, MS Access, MySQL**
- DBMS dirancang untuk dapat melakukan utilisasi dan mengelola koleksi data dalam jumlah yang besar. DBMS dapat melakukan manipulasi data secara lebih mudah.

Database Management System (DBMS)

- Sebelum adanya DBMS maka data pada umumnya disimpan dalam bentuk **flatfile**, yaitu **file teks** yang ada pada sistem operasi
- Sampai sekarangpun masih ada aplikasi yang menyimpan data dalam bentuk flat secara langsung.
- Menyimpan data dalam bentuk flat file mempunyai kelebihan dan kekurangan. Penyimpanan dalam bentuk ini akan mempunyai manfaat yang optimal jika ukuran filenya relatif kecil, seperti file passwd. File passwd pada umumnya hanya digunakan untuk menyimpan nama yang jumlahnya tidak lebih dari 1000 orang.

Database Management System (DBMS)

- Selain dalam bentuk flat file, penyimpanan data juga dapat dilakukan dengan menggunakan program bantu seperti **spreadsheet**. Penggunaan software ini memperbaiki beberapa kelemahan dari flat file, seperti bertambahnya kecepatan dalam pengolahan data. Namun demikian metode ini masih memiliki banyak kelemahan, diantaranya adalah masalah manajemen dan keamanan data yang masih kurang.

Database Management System (DBMS)

- Database Applications:
 - Banking: all transactions
 - Airlines: reservations, schedules
 - Universities: registration, grades
 - Sales: customers, products, purchases
 - Manufacturing: production, inventory, orders, supply chain
 - Human resources: employee records, salaries, tax deductions
- Databases touch all aspects of our lives

Purpose of Database System

- In the early days, database applications were built on top of file systems
- Drawbacks of using file systems to store data:
 - **Data redundancy and inconsistency**
 - Multiple file formats, duplication of information in different files
 - **Difficulty in accessing data**
 - Need to write a new program to carry out each new task
 - **Data isolation** — multiple files and formats
 - **Integrity problems**
 - Integrity constraints (e.g. account balance > 0) become part of program code
 - Hard to add new constraints or change existing ones

Purpose of Database Systems (Cont.)

- Drawbacks of using file systems (cont.)
 - **Atomicity of updates**
 - Failures may leave database in an inconsistent state with partial updates carried out
 - E.g. transfer of funds from one account to another should either complete or not happen at all
 - **Concurrent access by multiple users**
 - Concurrent accessed needed for performance
 - Uncontrolled concurrent accesses can lead to inconsistencies
 - E.g. two people reading a balance and updating it at the same time
 - **Security problems**
- Database systems offer solutions to all the above problems

Levels of Abstraction

- **Physical level** describes how a record (e.g., customer) is stored.
- **Logical level:** describes data stored in database, and the relationships among the data.

type customer = **record**

name : string;

street : string;

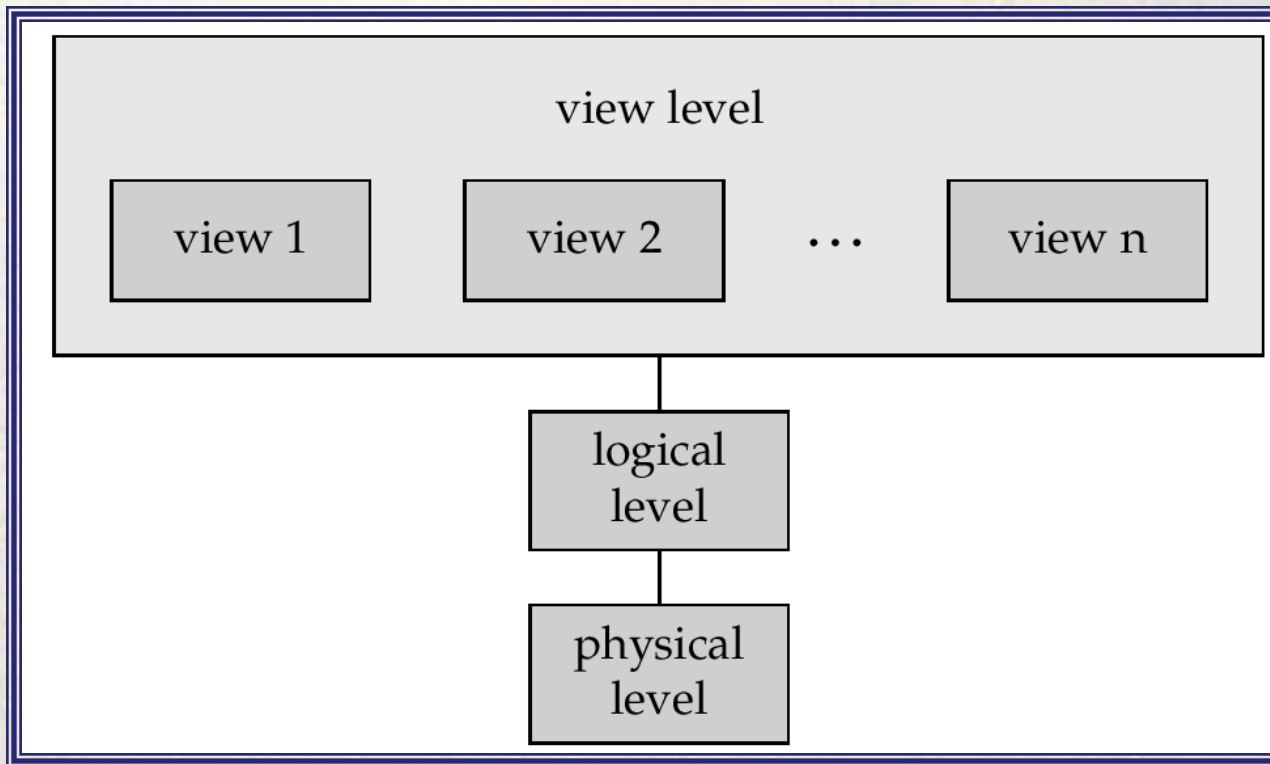
city : integer;

end;

- **View level:** application programs hide details of data types. Views can also hide information (e.g., salary) for security purposes.

View of Data

An architecture for a database system



Instances and Schemas

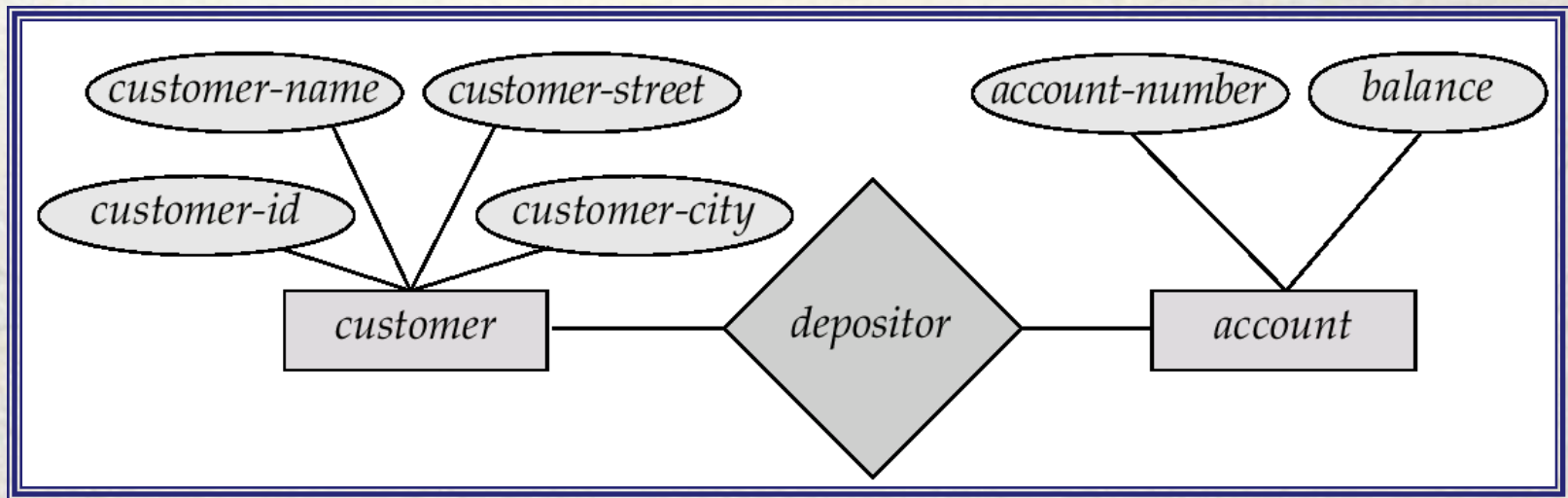
- Similar to types and variables in programming languages
- **Schema** – the logical structure of the database
 - e.g., the database consists of information about a set of customers and accounts and the relationship between them)
 - Analogous to type information of a variable in a program
 - **Physical schema**: database design at the physical level
 - **Logical schema**: database design at the logical level
- **Instance** – the actual content of the database at a particular point in time
 - Analogous to the value of a variable
- **Physical Data Independence** – the ability to modify the physical schema without changing the logical schema
 - Applications depend on the logical schema
 - In general, the interfaces between the various levels and components should be well defined so that changes in some parts do not seriously influence others.

Data Models

- A collection of tools for describing
 - data
 - data relationships
 - data semantics
 - data constraints
- Entity-Relationship model
- Relational model
- Other models:
 - object-oriented model
 - semi-structured data models
 - Older models: network model and hierarchical model

Entity-Relationship Model

Example of schema in the entity-relationship model

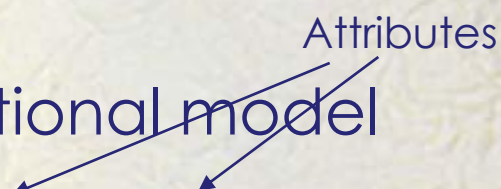


Entity Relationship Model (Cont.)

- E-R model of real world
 - Entities (objects)
 - E.g. customers, accounts, bank branch
 - Relationships between entities
 - E.g. Account A-101 is held by customer Johnson
 - Relationship set *depositor* associates customers with accounts
- Widely used for database design
 - Database design in E-R model usually converted to design in the relational model (coming up next) which is used for storage and processing

Relational Model

- Example of tabular data in the relational model



<i>Customer-id</i>	<i>customer-name</i>	<i>customer-street</i>	<i>customer-city</i>	<i>account-number</i>
192-83-7465	Johnson	Alma	Palo Alto	A-101
019-28-3746	Smith	North	Rye	A-215
192-83-7465	Johnson	Alma	Palo Alto	A-201
321-12-3123	Jones	Main	Harrison	A-217
019-28-3746	Smith	North	Rye	A-201

A Sample Relational

<i>customer-id</i>	<i>customer-name</i>	<i>customer-street</i>	<i>customer-city</i>
192-83-7465	Johnson	12 Alma St.	Palo Alto
019-28-3746	Smith	4 North St.	Rye
677-89-9011	Hayes	3 Main St.	Harrison
182-73-6091	Turner	123 Putnam Ave.	Stamford
321-12-3123	Jones	100 Main St.	Harrison
336-66-9999	Lindsay	175 Park Ave.	Pittsfield
019-28-3746	Smith	72 North St.	Rye

(a) The *customer* table

<i>account-number</i>	<i>balance</i>
A-101	500
A-215	700
A-102	400
A-305	350
A-201	900
A-217	750
A-222	700

(b) The *account* table

<i>customer-id</i>	<i>account-number</i>
192-83-7465	A-101
192-83-7465	A-201
019-28-3746	A-215
677-89-9011	A-102
182-73-6091	A-305
321-12-3123	A-217
336-66-9999	A-222
019-28-3746	A-201

(c) The *depositor* table

Data Definition Language (DDL)

- Specification notation for defining the database schema

- E.g.

- create table** *account* (
 account-number **char**(10),
 balance **integer**)

- DDL compiler generates a set of tables stored in a *data dictionary*
- Data dictionary contains metadata (i.e., data about data)
 - database schema
 - *Data storage and definition language*
 - language in which the storage structure and access methods used by the database system are specified
 - Usually an extension of the data definition language

Data Manipulation Language (DML)

- Language for accessing and manipulating the data organized by the appropriate data model
 - DML also known as query language
- Two classes of languages
 - Procedural – user specifies what data is required and how to get those data
 - Nonprocedural – user specifies what data is required without specifying how to get those data
- SQL is the most widely used query language

SQL

- SQL: widely used non-procedural language
 - E.g. find the name of the customer with customer-id 192-83-7465

```
select customer.customer-name
from customer
where customer.customer-id = '192-83-7465'
```
 - E.g. find the balances of all accounts held by the customer with customer-id 192-83-7465

```
select account.balance
from depositor, account
where depositor.customer-id = '192-83-7465' and
depositor.account-number = account.account-
number
```
- Application programs generally access databases through one of
 - Language extensions to allow embedded SQL
 - Application program interface (e.g. ODBC/JDBC) which allow SQL queries to be sent to a database