Getting Started with Databases

Session 1

Database for Big Data Analytics

Siti Mariyah, Ph.D.

Poilteknik Statistika STIS

Email: sitimariyah@stis.ac.id

Github: https://github.com/diahnuri

Session 1 – Getting Started with Databases

- Goal: Understand what databases are and why they matter in big data
- Content:
  - Some brief definitions and common use of database systems
  - Components of DBMS Environment
  - Data Model and Conceptual Modelling
  - How tables are used to represent data
  - Relational Model Terminology
  - Database roles in analytics pipeline
- Hands-on:
  - Installing/connecting to a database (e.g. pgAdmin or MySQL)
  - Connect to the provided database (via pgAdmin/Workbench/CLI)

What is Database?

- Database
  - A collection of related data
- Database Management System (DBMS)
  - A software that manages and control accesses to the database
- Database Application
  - A program that interacts with the database at some point in its execution
- Database System
  - A collection of application programs that interact with the database

Examples of Database Applications

- Purchases from the supermarket
  - Bar code reader → app prog → DB (update stock number)
- Purchases using your credit card
  - Card reader → app prog→ customer DB & stolen DB
- Booking a holiday at the travel agents
  - Ensure not to overbook the seat
- Using the local library
  - Bar code reader → app prog→ book DB (borrow & return)

File-based Systems

- Collection of application programs that perform services for the end users (e.g. reports).
- Each program defines and manages its own data.

File-based Systems

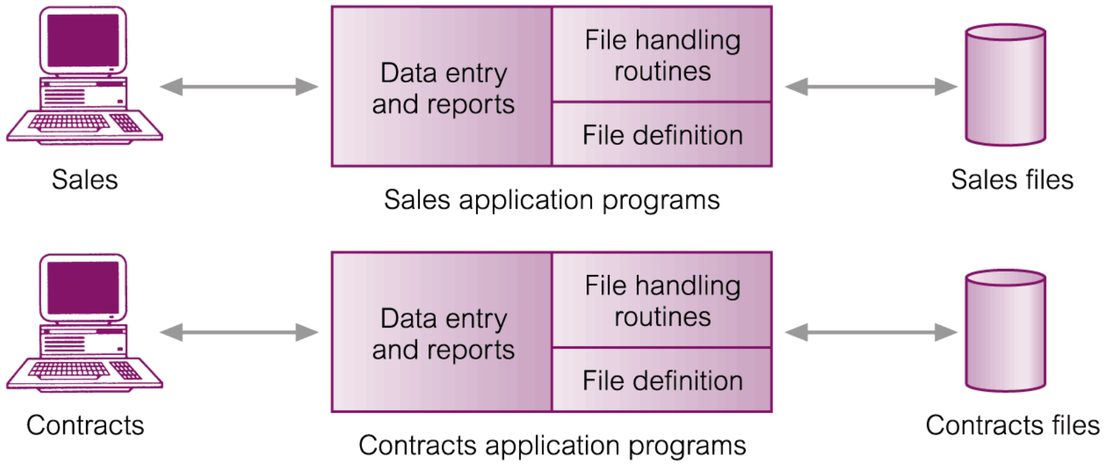


Figure 1.5
File-based processing.

Sales Files

PropertyForRent (propertyNo, street, city, postcode, type, rooms, rent, ownerNo)
PrivateOwner (ownerNo, fName, lName, address, telNo)
Client (clientNo, fName, lName, address, telNo, prefType, maxRent)

Contracts Files

Lease (leaseNo, propertyNo, clientNo, rent, paymentMethod, deposit, paid, rentStart, rentFinish, duration)
PropertyForRent (propertyNo, street, city, postcode, rent)
Client (clientNo, fName, lName, address, telNo)

Limitations of File-based Approach

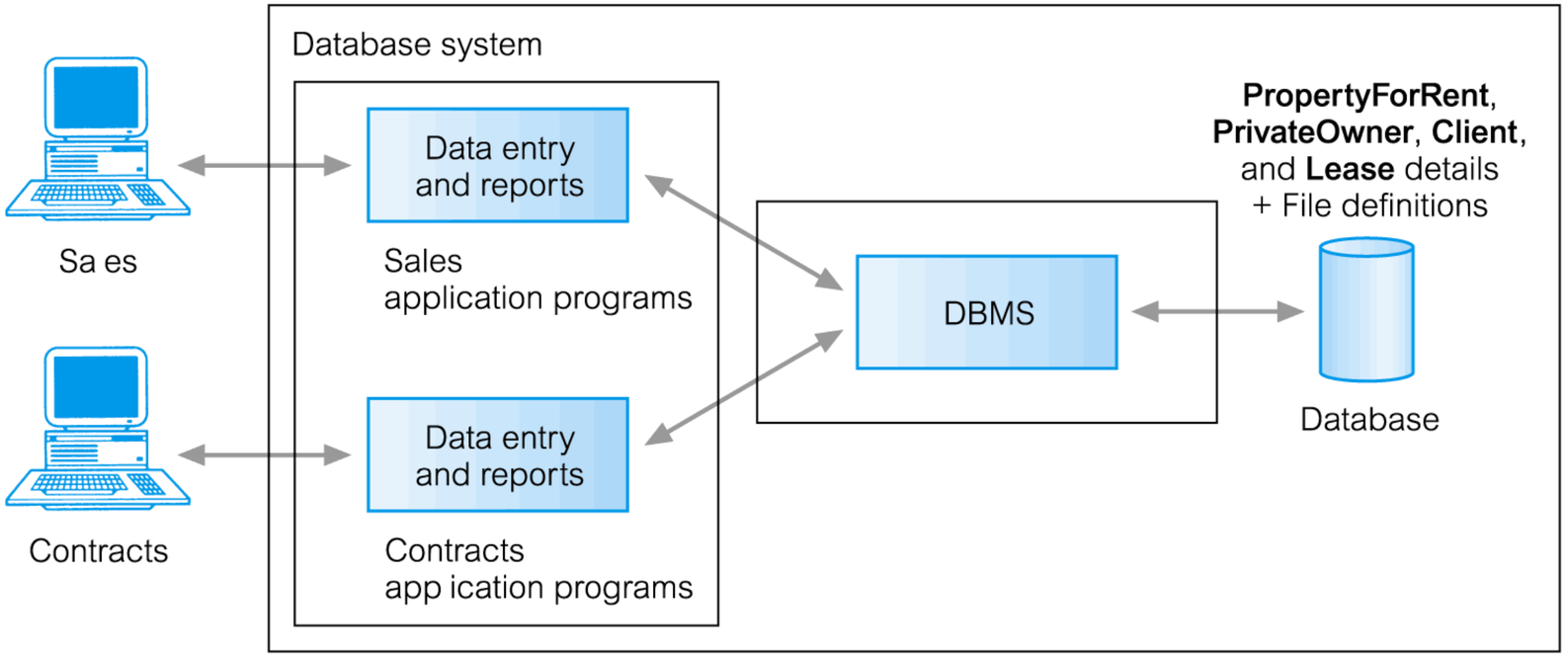
- Separation and isolation of data
  - Each program maintains its own set of data.
  - Users of one program may be unaware of potentially useful data held by other programs.
- Duplication of data
  - Same data is held by different programs.
  - Wasted space and potentially different values and/or different formats for the same item.
  - It costs time & money to enter data more than once.
  - Loss of data integrity.
- Fixed Queries of application programs
  - Programs are written to satisfy particular functions.
  - Any new requirement needs a new program.
- Program – Data dependence
  - File structure is defined in the program code.
  - It is difficult to change the file structure.
  - Need to change both the file and accessing programs.
- Incompatible file formats
  - Programs are written in different languages (e.g., COBOL & C) and so cannot easily access each other's files.

Database

- Definition: shared collection of logically related data and a description of this data, designed to meet the information needs of an organization.
  - Shared large repository of data with metadata (self-describing).
  - This enables *program–data independence*.
- Logically related data comprises entities, attributes, and relationships of an organization’s information.
  - **Entities**: branch and staff.
  - **Property**: branchNo and staffNo.
  - **Relationship**: a branch has staff.

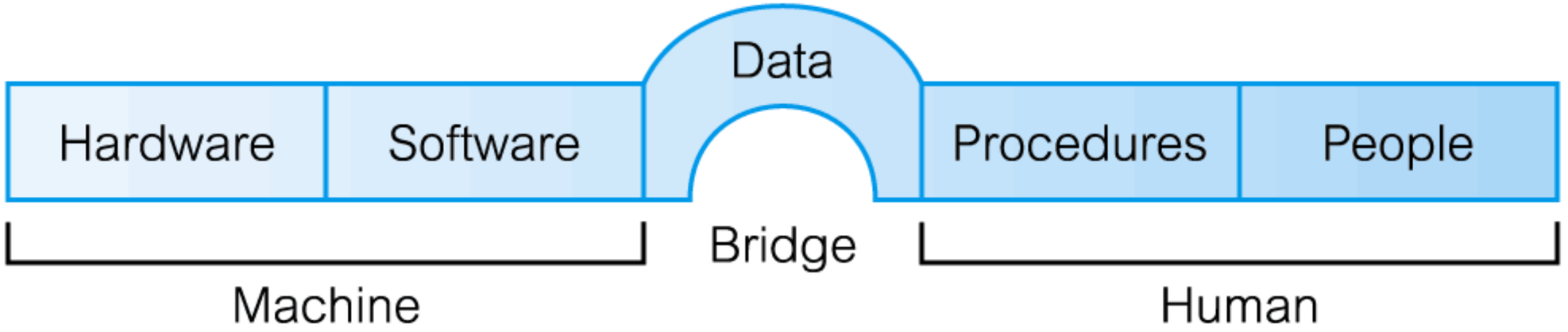
Database Management System (DBMS)

- A software system that enables users to *define*, *create*, *maintain*, and *control* access to the database.
- (Database) application program
  - A computer program that interacts with database by issuing an appropriate request (SQL statement) to the DBMS.



PropertyForRent (propertyNo, street, city, postcode, type, rooms, rent, ownerNo)
PrivateOwner (ownerNo, fName, lName, address, telNo)
Client (clientNo, fName, lName, address, telNo, prefType, maxRent)
Lease (leaseNo, propertyNo, clientNo, paymentMethod, deposit, paid, rentStart, rentF nish)

Component of DBMS Environment



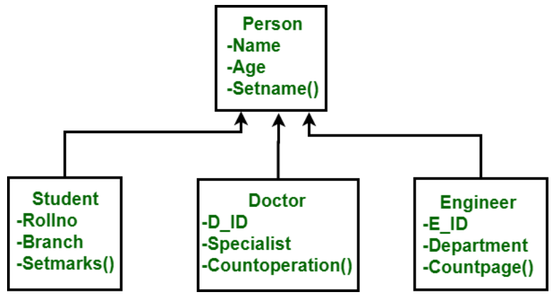
- Hardware
  - Can range from a PC to a network of computers.
  - Client-server architecture (backend & frontend).
- Software
  - DBMS, operating system, network software (if necessary) and the application programs .
- Data
  - Operational data used by the organization and a description of this data called the schema.
- Procedures
  - Instructions and rules that should be applied to the design and use of the database and DBMS.: Log in, start/stop DBMS, backup etc.
- People

Data Model

Purpose: to represent data in an understandable way.

Object-based

Entity: a distinct object that is to be represented in the db.
Attribute: a property of an entity that we wish to record.
Relationship: an association between entities.



Record-based

The DB consists of a number of fixed-format records possibly of differing types.

Relational Data Model

Data and relationships are represented as tables; each has a number of collums with a unique name.

Branch

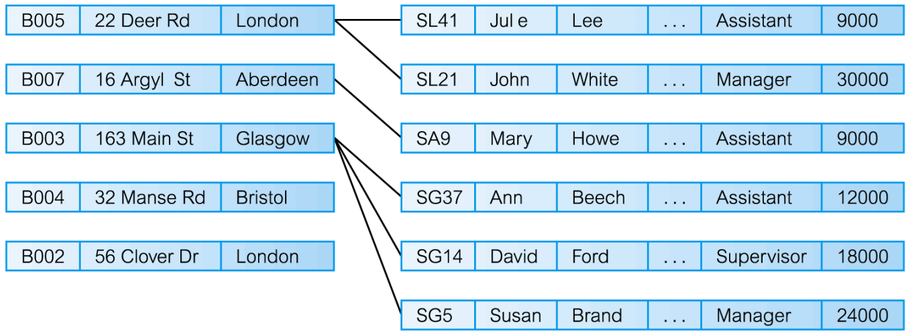
branchNo	street	city	postCode
B005	22 Deer Rd	London	SW1 4EH
B007	16 Argyll St	Aberdeen	AB2 3SU
B003	163 Main St	Glasgow	G11 9QX
B004	32 Manse Rd	Bristol	BS99 1NZ
B002	56 Clover Dr	London	NW10 6EU

Staff

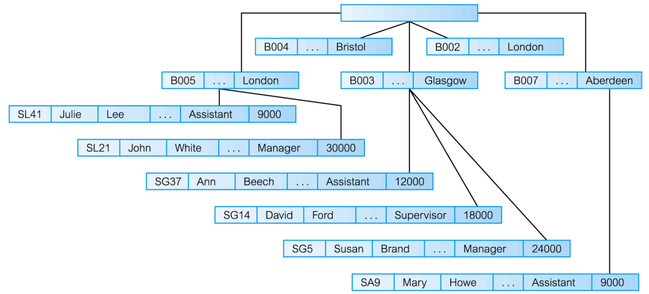
staffNo	fName	lName	position	sex	DOB	salary	branchNo
SL21	John	White	Manager	M	1-Oct-45	30000	B005
SG37	Ann	Beech	Assistant	F	10-Nov-60	12000	B003
SG14	David	Ford	Supervisor	M	24-Mar-58	18000	B003
SA9	Mary	Howe	Assistant	F	19-Feb-70	9000	B007
SG5	Susan	Brand	Manager	F	3-Jun-40	24000	B003
SL41	Julie	Lee	Assistant	F	13-Jun-65	9000	B005

Network Data Mode

Data and relationships are represented as a collection of records and sets, respectively.



Hierarchical Data Model



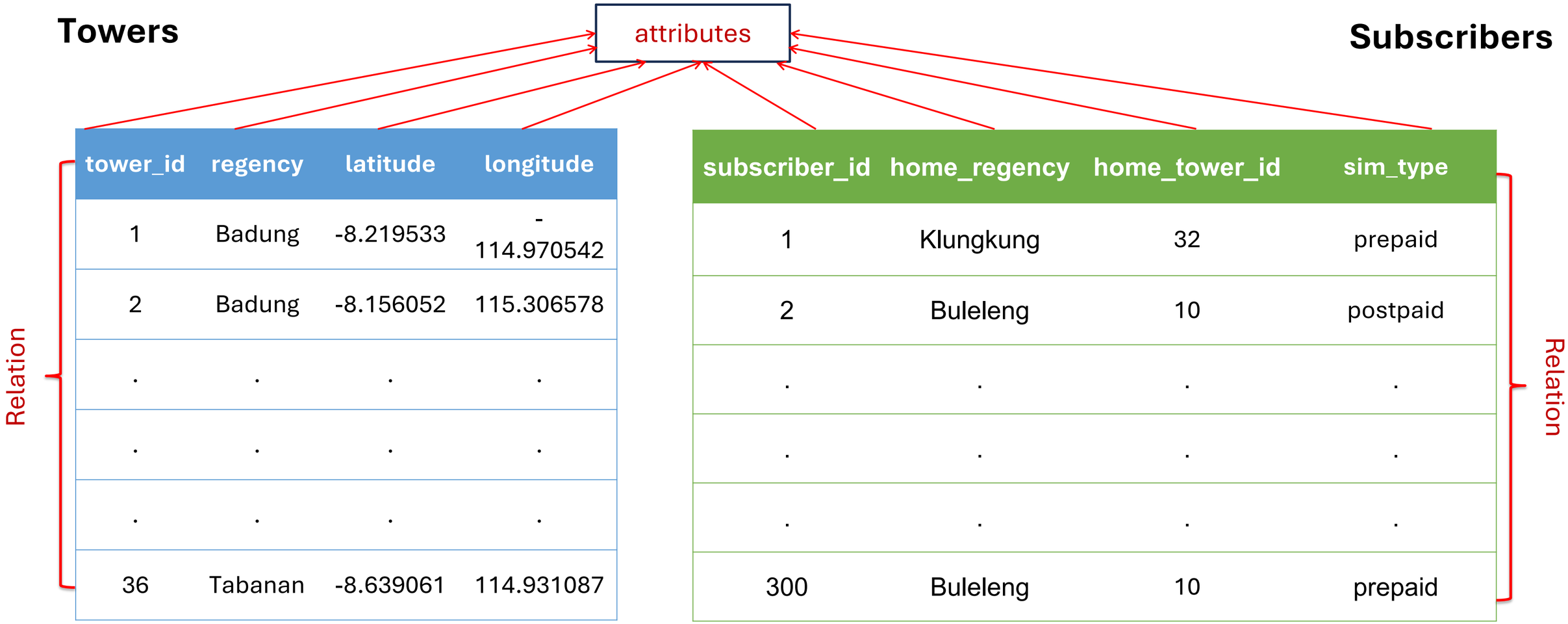
Relational Model Terminology

- A relation → a table with columns and rows.
  - Only applies to logical structure of the database, not the physical structure.
  - Hence, the external and conceptual levels.
- Attribute → a named column of a relation.
  - Used to hold information about objects to be represented in the database.
- Domain → the set of allowable values for one or more attributes.
  - Every attribute in a relation is defined on a domain.
  - May be distinct for each attribute, or two or more attributes may be defined on the same domain.
- Tuple → a row of a relation.
  - Can appear in any order & the relation is still the same
  - It is called as extension/state of a relation

Relational Model Terminology

- Intension → relation structure with domain and any other restrictions on possible values
- Degree → number of attributes in a relation
  - Unary relation: a relation with 1 attribute
  - Binary relation: a relation with 2 attributes
  - Ternary relation: a relation with 3 attributes
  - N-ary relation: a relation with more than 3 attributes
- Cardinality is the number of tuples in a relation.
- Relational Database is a collection of normalized relations with distinct relation names.

Instances of Towers and Subscribers Relations



tower_id	regency	latitude	longitude
1	Badung	-8.219533	-114.970542
2	Badung	-8.156052	115.306578
.	.	.	.
.	.	.	.
.	.	.	.
36	Tabanan	-8.639061	114.931087

subscriber_id	home_regency	home_tower_id	sim_type
1	Klungkung	32	prepaid
2	Buleleng	10	postpaid
.	.	.	.
.	.	.	.
.	.	.	.
300	Buleleng	10	prepaid

Examples of Attribute Domains

Attributes	Domain Name	Meaning	Domain Definition
tower_id	towerID	The set of all possible tower id	Integer number
regency	regency	All regencies in Indonesia	Character: size 100
latitude	latitude	Range of latitude values covering regency area	Float number
longitude	longitude	Range of longitude values covering regency area	Float number

Properties of Relations

- Relation name is distinct from all other relation names in relational schema.
- Each cell of relation contains exactly one atomic (single) value.
- Each attribute has a distinct name.
- Values of an attribute are all from the same domain.
- Each tuple is distinct; there are no duplicate tuples.
- Order of attributes has no significance.
- Order of tuples has no significance, theoretically.

Relational Keys

- Superkey
  - An attribute, or set of attributes, that uniquely identifies a tuple within a relation.
- Candidate Key
  - Superkey (K) such that no proper subset is a superkey within the relation.
  - In each tuple of R, values of K uniquely identify that tuple (*uniqueness*).
  - No proper subset of K has the uniqueness property (*irreducibility*).
- Primary Key
  - Candidate key selected to identify tuples uniquely within relation.
- Alternate Keys
  - Candidate keys that are not selected to be primary key.
- Foreign Key
  - Attribute, or set of attributes, within one relation that matches candidate key of some (possibly same) relation.

Instances of Towers and Subscribers Relations

Towers

Primary Key

Foreign Key

Subscribers

Relation

tower_id	regency	latitude	longitude
1	Badung	-8.219533	-114.970542
2	Badung	-8.156052	115.306578
.	.	.	.
.	.	.	.
.	.	.	.
36	Tabanan	-8.639061	114.931087

subscriber_id	home_regency	home_tower_id	sim_type
1	Klungkung	32	prepaid
2	Buleleng	10	postpaid
.	.	.	.
.	.	.	.
.	.	.	.
300	Buleleng	10	prepaid

Relation

Regional Hub
On Big Data and
Data Science
for Asia and The Pacific

Hands-on

hub.bps.go.id

Install DBMS PgAdmin on Windows

- To install pgAdmin, the graphical user interface (GUI) for PostgreSQL, you will first need to download and install PostgreSQL itself.
- https://www.enterprisedb.com/downloads/postgres-postgresql-downloads

Installation on Windows

- Download the installer: Go to the official PostgreSQL website and download the Windows installer from EnterpriseDB (EDB).
- Run the installer: Launch the downloaded executable file. The setup wizard will guide you through the process.
- Choose components: During the installation, you will be prompted to select the components to install. Ensure that both PostgreSQL Server and pgAdmin 4 are checked.
- Set a password: Create a password for the default postgres database superuser. You will need this password to log in to pgAdmin and manage your database.
- Finish installation: Complete the wizard. Once finished, you can find pgAdmin 4 in your Start menu under the PostgreSQL folder.

Install DBMS PgAdmin on Mac

- To install pgAdmin, the graphical user interface (GUI) for PostgreSQL, you will first need to download and install PostgreSQL itself.
- https://www.postgresql.org/download/macosx/

Installation on Mac

- Download the installer: Visit the PostgreSQL download page and get the graphical installer for macOS from EDB.
- Run the installer: Open the downloaded .dmg file and follow the on-screen instructions.
- Select components: Just like on Windows, select both the PostgreSQL Server and pgAdmin 4 components during setup.
- Complete the installation: After the installation is finished, you can find pgAdmin 4 in your Applications folder.
  Alternatively, advanced users can install PostgreSQL and pgAdmin 4 using Homebrew.

Thank You

Email: sitimariyah@stis.ac.id
Github: https://github.com/diahnuri