

Database Usage Patterns

Data engineers and database administrators



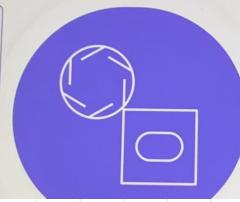
Access databases for administrative tasks

Include:

- Creation and management of database objects
- Access controls
- Monitoring and performance tuning

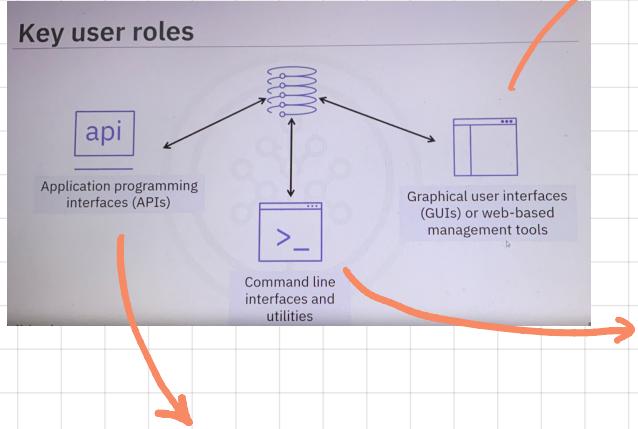
GUIs

- Are common means for interacting with databases
- Includes graphical, web, and mobile

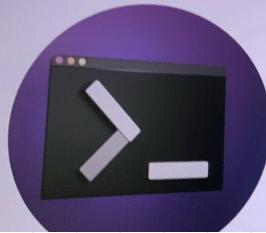


- Users may opt for third-party alternatives
- Examples: Oracle SQL Developer manages Oracle databases

Key user roles



Command line interfaces and utilities



- Are still prevalent
- Requires proficiency in usage
- Allows users to:
 - Issue database commands directly
 - Utilize interactive shells for command-line interactions

APIs



Object relational mapping frameworks



ActiveRecord in Ruby applications



Hibernate in Java



Django in Python



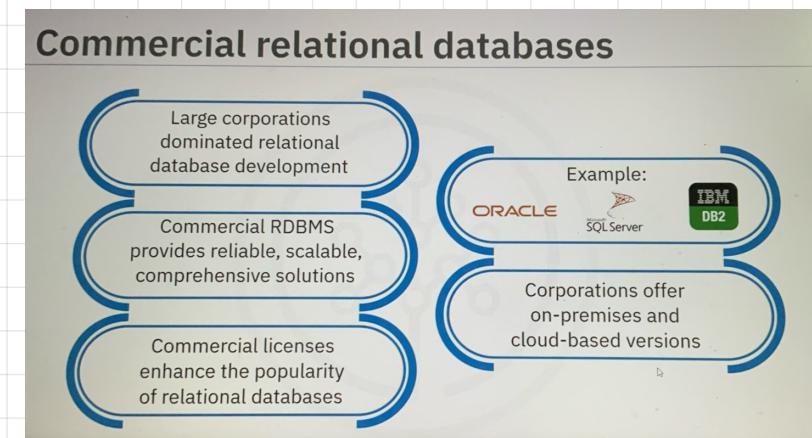
Sequelize in JavaScript



Entity Framework in .NET

Introduction to Relational Database

Commercial relational databases



Most popular relational databases



Db2

Db2 Features

- AI-powered functionality:
 - Machine learning algorithms
 - Column store
 - Data skipping
- Common SQL Engine
- Support for all data types
- High availability and disaster recovery
- Scalability
- Table partitioning

Mod 2.

Creating Table and Loading Data:

Types of SQL Statements (DDL vs DML)

Types of SQL Statements - DDL

- SQL Statement types: DDL and DML
- DDL (Data Definition Language) statements:
 - Define, change, or drop data
- Common DDL:
 - CREATE
 - ALTER {
Add
Drop
Modify}
 - TRUNCATE {
Delete data table, but not data itself}
 - DROP

Actions with: Entity : tables

Att.: columns

Tuples: rows + data values

Types of SQL Statements - DML

- DML (Data Manipulation Language) statements:
 - Read and modify data *refer to*
 - CRUD operations (Create, Read, Update & Delete rows)
- Common DML:
 - INSERT
 - SELECT
 - UPDATE
 - DELETE

ALTER, DROP and TRUNCATE

Alter :

ALTER TABLE ... ADD COLUMN

- Add or remove columns
- Modify the data type of columns
- Add or remove keys
- Add or remove constraints

```
ALTER TABLE <table_name>
  ADD COLUMN <column_name_1> datatype
  ...
  ADD COLUMN <column_name_n> datatype;
```

ALTER TABLE ... ADD COLUMN

```
ALTER TABLE author
  ADD COLUMN telephone_number BIGINT;
```

author_id	lastname	firstname	email	city	country	telephone number
1001	Thomas	John	johnt@...	New York	USA	555 1111
1002	James	Alice	alicej@...	Seattle	USA	555 1112
1003	Wells	Steve	stevew:@...	Montreal	Canada	555 2222
1004	Kumar	Santosh	kumars@...	London	UK	555 3333

to Data type

ALTER TABLE ... ALTER COLUMN

```
ALTER TABLE author
  ALTER COLUMN telephone_number SET DATA TYPE
    CHAR(20) set to CHAR(20) int('1')
```

author_id	lastname	firstname	email	city	country	telephone number
1001	Thomas	John	johnt@...	New York	USA	555-1111
1002	James	Alice	alicej@...	Seattle	USA	555-1112
1003	Wells	Steve	stevew:@...	Montreal	Canada	555-2222
1004	Kumar	Santosh	kumars@...	London	UK	555-3333

int : 31-167769168 Data type

TRUNCATE TABLE => Delete all rows data

```
TRUNCATE TABLE author
IMMEDIATE;
```

author_id	lastname	firstname	email	city	country	
1001	Thomas	John	johnt@...	New York	USA	
1002	James	Alice	alicej@...	Seattle	USA	
1003	Wells	Steve	stevew:@...	Montreal	Canada	
1004	Kumar	Santosh	kumars@...	London	UK	

TRUNCATE TABLE

```
TRUNCATE TABLE author
IMMEDIATE;
```

author_id	lastname	firstname	email	city	country	

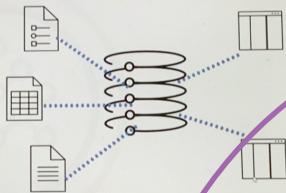
Data Movement tools and utilities

Scenarios for data movement

The need to move data in and out.

Reasons:

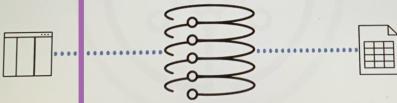
- Initially populate the entire database
- Create a copy for development and testing
- Create a snapshot for disaster recovery
- Create a new table from an external source/file
- Add data to an existing table



Data movement tools and utilities

- Multiple methods and file types supported
- Data movement tool categories:
 - Backup and restore
 - Import and export
 - Load

{ this class has classified for three categories



Backup and restore



- Backup creates a file for the entire database
- Restore duplicates the database precisely
- Preserves all database objects and their data:
 - Schemas, tables, views
 - User-defined data types, functions, stored procedures
 - Constraints, triggers, security, relationships, and data

Import and export

- Import inserts data into table from file
- Export saves table data into a file

Import and export

Operations available using different interfaces



Import and export file formats

DEL	<ul style="list-style-type: none">• Utilizes special character delimiters• A common approach to storing data
ASC	<ul style="list-style-type: none">• Imports or loads data from applications• Generates flat text files with aligned columns
PC/IXF	<ul style="list-style-type: none">• The preferred method for data exchange• Presents a structured description of the database tab
JSON	<ul style="list-style-type: none">• Popular web services• Databases and tools now import/export JSON data

Load utilities

- Alternative to the import utility
- Faster than the import utility
- Doesn't perform as many checks
- Bypasses database logging
- Preferred option for loading large data sets
- Initiate from command line / API / Visual tool

```
db2 load from filename of fileformat messages messagesfile import_mode  
into table copy yes/no use tsm data buffer pages
```

Designing Keys, Indexes, and Constraints

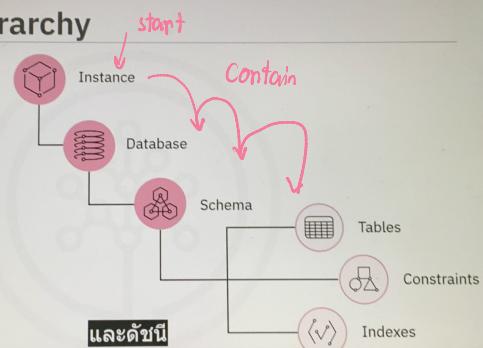
Databases Objects and Hierarchy (including schema)

Database hierarchy

- RDBMSs contain many objects
- Enables to:
 - Manage security
 - Maintenance
 - Accessibility
- Provides an overview of the RDBMS structure

Database hierarchy

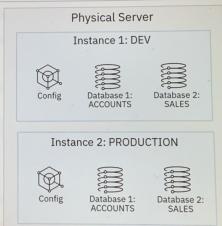
- Instance
- Database
- Schema
 - Logical grouping of objects
 - Define how to name database objects
- Database objects



1.

Instances

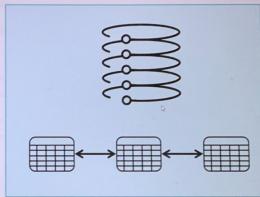
- Instance:
 - Logical boundary for databases
 - Organize objects
 - Set configuration parameters
- Provides unique database server environment
- Allows isolation between databases
- Not all RDBMSs use instances



2.

Relational databases

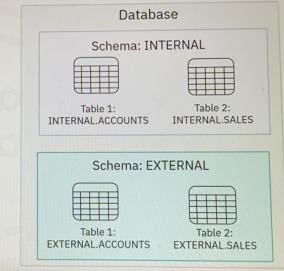
- Stores, manages, and provides access to data
- Contains objects like tables, views, indexes
- Uses related tables to reduce data redundancy
- Distributed across multiple systems



3.

Schemas

- Organize database objects
- Include various objects
- Assign it to a schema
- Include schema name
- Default schema is the user schema
- Provides a naming context
- Enables you to distinguish objects



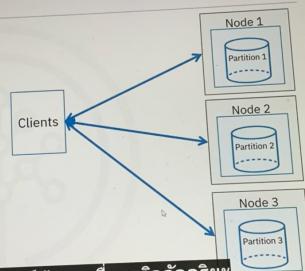
Schemas

- Many RDBMSs use a specialized schema
- Tables in a system schema store:
 - Lists of database users
 - Access permissions
 - Information about the indexes
 - Details of any database partitions
 - User-defined data types

4.

Database partitions

- Data managed across multiple partitions
- Split tables that contain very large quantities of data
- Hold a subset of the data
- Common in data warehousing



Database objects

Common database objects:

- Tables
- Constraints
- Indexes
- Views
- Aliases

Create using Graphical tools,
scripting, APIs, SQL



Tables



Constraints



Indexes

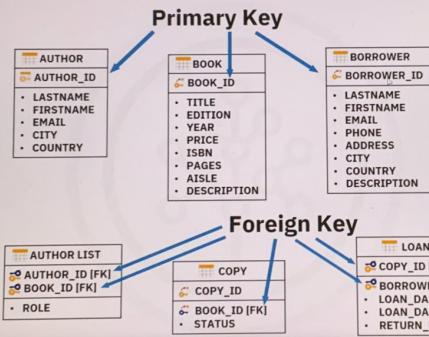
Primary key & Foreign key

Creating primary keys

```
CREATE TABLE book(
    book_id INT NOT NULL,
    .
    .
    pub_id INT NULL,
    PRIMARY KEY(book_id));
```

```
ALTER TABLE book
ADD PRIMARY KEY(book_id, ISBN);
```

What is a foreign key?



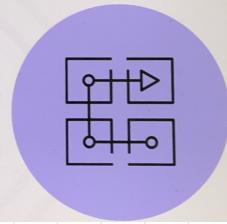
Creating foreign keys

```
CREATE TABLE copy(
    copy_id INT NOT NULL,
    book_id INT NULL,
    .
    .
    CONSTRAINT fk_copy_book FOREIGN KEY(book_id)
    REFERENCES book(book_id)
);
```

Normalization

What is normalization?

- Data duplication leads to inconsistencies
- Normalization reduces data duplication
- Increases speed of transactions
- Improves the integrity of data
- Normalizes each table
- Most used:
 - First normal form
 - Second normal form
 - Third normal form



First normal form

Each row must be unique

Each cell must contain only a single value

Also called 1NF

1NF

First normal form

Book_id	Title	Format	Author_name
101	Lean Software Development	Paperback	Mary Poppendieck
201	Facing the Intelligence Explosion	Paperback	David Robson
301	Scala in Action	Hardback	Yehuda Katz
401	Patterns of Software	Paperback	Mary Poppendieck
501	Anatomy of LISP	Paperback	Eric Redmond

First normal form

Book_id	Title	Format	Author_name
101	Lean Software Development	Paperback	Mary Poppendieck
201	Facing the Intelligence Explosion	Paperback	David Robson
301	Scala in Action	Hardback	Yehuda Katz
401	Patterns of Software	Paperback	Mary Poppendieck
501	Anatomy of LISP	Paperback	Eric Redmond

Second normal form

Database must be in the first normal form

Create separate tables for sets of values

Also called 2NF

2NF

Second normal form

Book_id	Title	Format	Author_name
101	Lean Software Development	Paperback	Mary Poppendieck
201	Facing the Intelligence Explosion	Paperback	David Robson
301	Scala in Action	Hardback	Yehuda Katz
401	Patterns of Software	Paperback	Mary Poppendieck
501	Anatomy of LISP	Hardback	Mary Poppendieck

Second normal form

Book_id (Primary Key)	Title	Author_name	Book_id (Foreign Key)	Format
101	Lean Software Development	Mary Poppendieck		Paperback
201	Facing the Intelligence Explosion	David Robson		Paperback
301	Scala in Action	Yehuda Katz		Hardback
401	Patterns of Software	Mary Poppendieck		Hardback
501	Anatomy of LISP	Eric Redmond		Paperback

Third normal form

Database must be in the first and second normal form

Eliminate columns that do not depend on the key

Also called 3NF

3NF

Third normal form

Book_id (Primary Key)	Title	Author_name	Publisher	Ships from
101	Lean Software Development	Mary Poppendieck	Tech Books	UK
201	Facing the Intelligence Explosion	David Robson	Amazing Books	US
301	Scala in Action	Yehuda Katz	Better Tech Books	India
401	Patterns of Software	Mary Poppendieck	Publisher 101	US

Wanna Book_id
Wanna Publisher
Wanna Shippin'

Third normal form

Book_id (Primary Key)	Title	Author_name	Pub_id	Publisher	Ships from
101	Lean Software Development	Mary Poppendieck	1	Tech Books	UK
201	Facing the Intelligence Explosion	David Robson	2	Amazing Books	US
301	Scala in Action	Yehuda Katz	3	Better Tech Books	India
401	Patterns of Software	Mary Poppendieck	4	Publisher 101	US

Normalization in OLTP and OLAP

- Online transactional processing (OLTP)
 - Involves frequent reading and writing of data
 - Normalizes data to 3NF
- Online analytical processing (OLAP)
 - Primarily deals with read-only data

Relational Model Constraints - Advanced

Constraints

Relational data model needs to adhere to rules using:

- Entity integrity constraint
- Referential integrity constraint
- Semantic integrity constraint
- Domain constraint
- Null constraint
- Check constraint

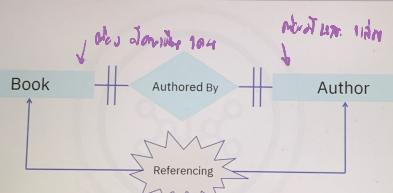
Entity integrity constraint

The primary key is a unique value that identifies each tuple (or row) in a table

AUTHOR

Author_ID [PK]	Lastname	Firstname	Email	City	Country
NULL	Chong	Raul	rfc@ibm.com	Toronto	CA
A2	Ahuja	Rav	ra@ibm.com	Toronto	CA
A3	Hakes	Ian	ih@ibm.com	Toronto	CA
NULL	Sharma	Neeraj	ns@ibm.com	Chennai	IN
A5	Perniu	Liviu	lp@univ.com	Transylvania	RO

Referential integrity constraint



Semantic integrity constraint

AUTHOR

Author_ID [PK]	Lastname	Firstname	Email	City	Country
Null	Chong	Raul	rfc@ibm.com	12(*)&^23	CA
A2	Ahuja	Rav	ra@ibm.com	Toronto	CA
A3	Hakes	Ian	ih@ibm.com	Toronto	CA
Null	Sharma	Neeraj	ns@ibm.com	Chennai	IN
A5	Perniu	Liviu	lp@univ.com	Transylvania	RO

Domain constraint

AUTHOR

Author_ID [PK]	Lastname	Firstname	Email	City	Country
NULL	Chong	Raul	rfc@ibm.com	Toronto	34 J
A2	Ahuja	Rav	ra@ibm.com	Toronto	CA
A3	Hakes	Ian	ih@ibm.com	Toronto	CA
NULL	Sharma	Neeraj	ns@ibm.com	Chennai	IN
A5	Perniu	Liviu	lp@univ.com	Transylvania	RO

Check constraint

Check constraint

Title	Edition	Year	Price	ISBN	Pages	Aisle	Description
Database Fundamentals	1	2010	24.99	978-0-9866283-1-1	300	DB-A02	Teaches you the fundamentals of databases
Getting started with DB2 Express-C	1	2010	24.99	978-0-9866283-5-1	280	DB-A01	Teaches you the essentials of DB2 using DB2 Express-C, the free version of DB2

Mod 3

MySQL : pass

PostgreSQL :

PostgreSQL: Introduction

- Open-source object-relational database management system
- Reliable and flexible
- Supports both relational and non-relational data types



suitable for OLTP, Data Analytic and Geographic information system

Connectivity tools

Psql <ul style="list-style-type: none">• A command-line interface• Text-based interactions	pgAdmin <ul style="list-style-type: none">• Used as an open-source graphical interface• Simplifies database management
Commercial options <ul style="list-style-type: none">• Navicat and Dbeaver• Provides graphical interfaces supporting PostgreSQL	Managed database services <ul style="list-style-type: none">• Amazon RDS for PostgreSQL• Streamlines database administration in a cloud environment

Using psql

- Used as the default command-line interface
- Facilitates connections through straightforward commands
- Specifies various parameters
- Transforms the terminal into an interactive shell
- Performs a range of tasks
- Uses features like auto-completion and syntax highlighting



Creating Databases and Loading Data in PostgreSQL

Using psql to create a database

```
CREATE DATABASE employees;
\connect employees;
CREATE TABLE employee_details (firstname VARCHAR(20), lastname
VARCHAR(20), startdate DATE, salary DECIMAL);
```

Column	Type	Collation	Nullable	Default
firstname	character varying(20)			
lastname	character varying(20)			
startdate	date			
salary	numeric			

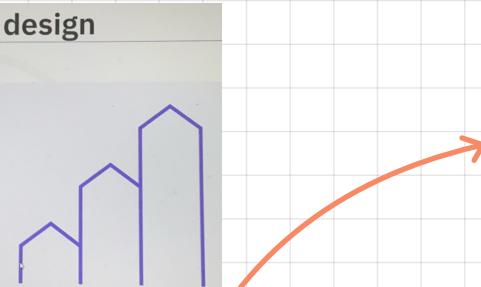
คุณสามารถใช้ PSQL เพื่อออกคำสั่งเพื่อสร้างและໂທ

Mod 4

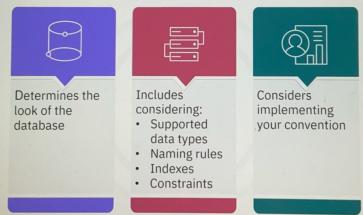
Approach to database Design (including ERD)

Importance of database design

- Crucial to the success of a project
- Contributes to:
 - Integrity of data
 - No redundancy
 - Performance
 - User satisfaction
- Avoids costly problems



Physical design



Database design process



Requirements analysis

The initial phase

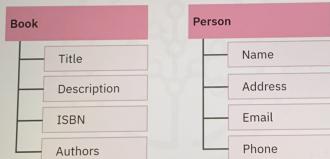
Involves:

- Working closely with stakeholders
- Analyzing real-world business information and policies

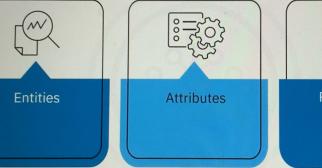
Identify the base objects

Requirements analysis

- Identify information associated with objects
- Example: Details of the book



Logical design



Logical design

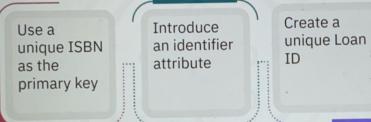
- Objects evolve into entities
- Ensure objects stand as entities
- Object characteristics become attributes

Book	Loan	Person
Title Description ISBN Authors	Book title Person first name Person last name Loan date Return date	First name Last name Street address City Email Phone

Considering entity attributes

- Includes consideration of entity attributes
 - Example: Storing address into parts
- Establishes a conceptual blueprint
- Ensures a robust foundation for physical design

Entity relationship management



Normalization

Normalization:

- Is the database design process
- Minimizes data redundancy

Online transaction processing (OLTP):

- Aim for the third normal form

Online analytical processing (OLAP):

- Prioritize denormalization

First Normal Form:

- Eliminate multiple author names in an attribute

Data acquisition

Reviewing existing data stores:

- Use it as a source of information

Determining current data usage

Gathering insights for potential improvements

Requirement Analysis Output



- Report
- Data diagram
- Presentation