- **ACID Properties**
- **Primary Key and Unique Key**
- **Foreign Key in RDBMS**

Research

by

Name: Islam Abdelraouf

Course: PHP Diploma

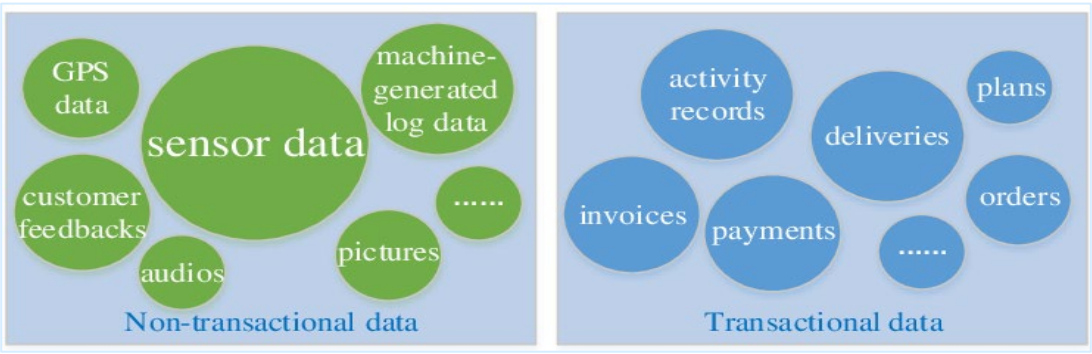
Date: 30-May-2025

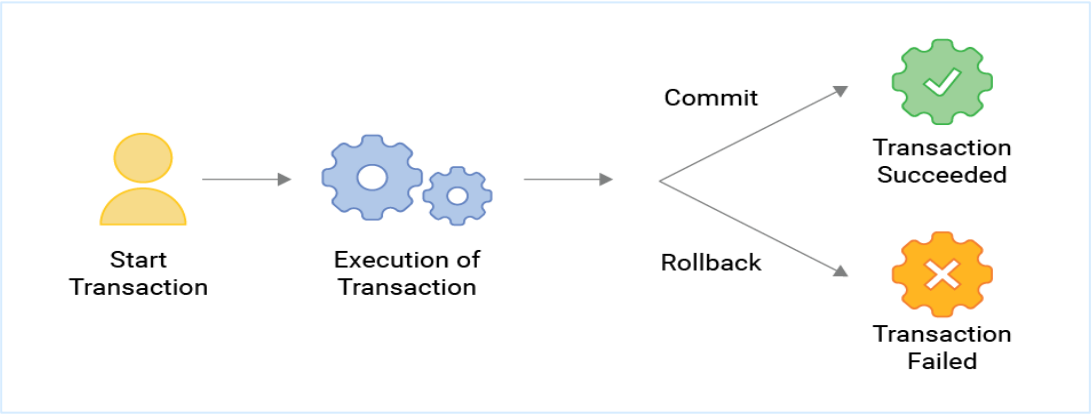
ACID Properties

ACID are four properties that must be satisfied when carrying out a database transaction operation. Database transaction is used to handle sensitive information e.g. invoices, money transactions, orders, or payments.

During transaction operations a set of steps must be executed against a database using DBMS, web application, or Saas service, to satisfy the ACID properties, to ensure sensitive information are assessed and fully updated permanently even if the system fails afterwards.

Transaction in a simple way is the operation of "ALL OR NONE" means all data must be updated to the database or none will be updated and rolling back action to take place in case of any step fails.





To execute this set of operations, four conditions have to be strictly committed and verified, they are called ACID properties:

ATOMICITY

All to be executed or none of them and rolling back to prevent partial updates.

Consistency

During all execution steps, all data integrity and relations constrains rules must be followed.

Isolation

Simultaneous transactions to be done separately without interference, bank accounts transactions.

Durability

To ensure that all transactions are carried out permanently even if the system fails afterwards

Primary key && Unique key

"PRIMARY KEY" and "UNIQUE" are both constrains that can be applied for columns to ensure respective data are not repeated.

However, primary key must be only one for each table, and "UNIQUE" key can be applied for as many columns as required.

Primary key doesn't allow duplicates or null values, and by default adds "NOT NULL" constraint, but "UNIQUE" key allows null values.

These both constrains could be added while creating the table or after creation using "ALTER" keyword.

CASE	EXAMPLE

Create table with one unique key	`CREATE TABLE users ( id INT UNIQUE NOT NULL, username VARCHAR(50) );`
	
	`CREATE TABLE users ( id INT NOT NULL, username VARCHAR(50), UNIQUE(id) );`

Create table, adding unique key later	`CREATE TABLE users ( id INT NOT NULL, username VARCHAR(50) );`
	
	`ALTER TABLE users ADD UNIQUE(id);`

CASE	EXAMPLE
A unique constraint can be named	`CREATE TABLE `users` ( id INT NOT NULL, username VARCHAR(50), CONSTRAINT constr_id(id) );`
Later same unique constraint can be removed	`ALTER TABLE `users` DROP CONSTRAINT constr_id;`
Create multi unique keys	`CREATE TABLE `users` ( user_guid INT NOT NULL, user_name VARCHAR(50) NOT NULL, user_pass VARCHAR(50) NOT NULL, user_age INT NOT NULL, CONSTRAINT uc_user(user_guid,user_age) );`
Create a primary key while creating the table	`CREATE TABLE `users` ( user_guid INT PRIMAARY KEY, user_name VARCHAR(50) NOT NULL, user_pass VARCHAR(50) NOT NULL, user_age INT NOT NULL, );` `CREATE TABLE `users` ( user_guid INT, user_name VARCHAR(50) NOT NULL, user_pass VARCHAR(50) NOT NULL, user_age INT NOT NULL, PRIMAARY KEY (user_guid) );`
Create a primary key while creating the table and assigning a name to it	`CREATE TABLE `users` ( user_guid INT, user_name VARCHAR(50) NOT NULL, user_pass VARCHAR(50) NOT NULL, user_age INT NOT NULL, CONSTRAINT pk_userID PRIMAARY KEY(user_guid) );`
Add primary key after creating the table	`ALTER TABLE `users` ADD PRIMAARY KEY (user_guid);`

CASE	EXAMPLE
Add primary key and assigning a name to it after creating the table	`ALTER TABLE `users` ADD CONSTRAINT pk_userID PRIMAARY KEY (user_guid);`
Delete the primary key	`ALTER TABLE `users` DROP PRIMAARY KEY;`

Foreign key in RDBM

The foreign key is a key that references to a primary key in another table, the table that has the foreign key is called a child table, and the table with the primary key is called a parent table.

A foreign key can be assigned to a field or to group of fields in a table, and must refers to a primary key in another table.

The foreign key can be created while creating the table, or after by using any of below methods:

- ALTER TABLE `table_name` ADD FOREIGN KEY (filed_name);

- ALTER TABLE `table_name` CONSTRAINT fk_name FOREIGN KEY (filed_name);

REFERENCES

wikipedia.org

wikipedia.org

budibase

w3schools.com

w3schools.com

w3schools.com