

# UJIAN AKHIR SEMESTER TAKE HOME

## SISTEM BASIS DATA

NIM : 1606238  
Nama : Muhamad Faisal

### 1. Integritas Data

- Pengertian Integritas Data

Integritas data adalah menjamin konsistensi data terhadap semua konstrain yang diberlakukan terhadap data tersebut, sehingga memberikan jaminan keabsahan data itu sendiri.

Tujuannya adalah menjaga data agar data yang tidak valid tidak masuk ke database. Menjaga konsistensi data pada relasi keterkaitan antar tabel. Integritas data dapat dikelompokkan sebagai berikut:

- Entity Integritas

Entity integritas mendefinisikan sebuah baris sebagai sebuah entitas yang unik untuk suatu tabel. Entity integritas memaksa integritas dari column atau primary key dari suatu tabel (melalui index, unique, constraints, primarykey) tidak boleh null.

- Domain integritas

Domain Integritas merupakan validasi dari masukan untuk sebuah kolom. Anda dapat memaksa domain integritas dengan membatasi tipe (melalui data types), format (melalui check constraints dan rules), atau range nilai-nilai yang mungkin (melalui Foreign Key Constraints, Check Constraints, Default Definitions dan rules).

- Refferential Integritas

Refferential integritas memastikan bahwa seluruh nilai dari foreign key cocok dengan nilai primary key yang dihubungkan.

- User Defined integritas

User defined integritas mengizinkan kita untuk menentukan spesifik business rules sendiri yang tidak sama pada kategori integrity yang lain.

- Jenis – Jenis Integritas Data

- Domain integrity (Integritas jangkauan nilai)

Tidak ada item data yang melanggar jangkauan nilai di tiap kolom data.

- Integrity Entity (Integritas Entitas)

Tidak ada baris data duplikat dalam satu tabel.

- Integrity Referential (Integritas acuan)

Menjaga relasi atau korespondensi antar tabel.

- Redudant Data Integrity (Integritas Data Antar Tabel)

Data di sebuah tabel tidak berulang di tabel lain.

- Bussiness Rule Integrity (Integritas Aturan Nyata)

Data di suatu tabel harus memiliki nilai yang layak dan dapat diterima ditempat sistem basis data itu diterapkan. Integritas jenis ini bersifat kasuistis, artinya sangat bergantung pada jenis data, lokasi, lingkungan, dan waktu penerapan basis data.

## 2. Transaksi

Transaksi adalah satu atau beberapa aksi program aplikasi yang mengakses atau mengubah isi basis data. Tujuannya untuk melindungi data pada database dari kehilangan data ataupun kerusakan data. Contohnya ketika kita mengambil uang dari sebuah atm dan terjadi kegagalan system maka semua transaksi harus berhenti dan kembali ke point sebelumnya.

Ada dua operasi penting dalam transaksi:

### a. Commit

Memberi tanda bahwa transaksi telah berhasil dan selesai sehingga update telah dibuat permanen.

### b. Rollback

Memberi tanda bahwa transaksi gagal sehingga transaksi akan dikembalikan kepada point sebelumnya atau dengan kata lain melakukan proses undo.

Ada 4 elemen transaksi

- a. Atomicity : Semua berhasil atau semua gagal.
- b. Consistency : Transaksi mempertahankan konsisten database.
- c. Isolation : Transaksi terisolasi satu dengan lainnya
- d. Durability : Setelah commit update harus survive di database.

### 3. Store Procedure

Store Procedure adalah objek yang tersimpan pada MySQL dan dapat digunakan untuk mengganti kumpulan perintah yang sering kita gunakan, seperti sejumlah row ke table lain dengan filter tertentu. Beberapa alasan penggunaan Store Procedure:

- Dijalankan dalam database sehingga performance lebih baik, network trafik lebih rendah.
  - Tools DBA sehingga lebih praktis.
  - Dapat digunakan oleh banyak aplikasi.
  - Dapat memanfaatkan tipe, fungsi yang disediakan Database MS.
- 
- Kelebihan Stored Procedures
    - Berbagi logic dengan aplikasi lainnya. Stored procedures merangkum fungsionalitas untuk memastikan bahwa akses data dan manipulasi koheren antara aplikasi yang berbeda.
    - Mengisolasi pengguna dari tabel data. Fitur ini memberi Anda kemampuan untuk memberikan akses ke stored procedures yang memanipulasi data namun tidak secara langsung ke tabel.
    - Berikan mekanisme keamanan. Jika Anda hanya dapat mengakses data menggunakan stored procedures yang telah ditentukan, tidak ada orang lain yang dapat menjalankan pernyataan SQL DELETE dan menghapus data Anda.
    - Untuk meningkatkan kinerja karena mengurangi lalu lintas jaringan. Dengan stored procedures, beberapa panggilan bisa disatukan menjadi satu.

- Kekurangan Stored Procedures
  - Peningkatan beban pada database server. Sebagian besar pekerjaan dilakukan di sisi server, dan kurang pada sisi client.
  - Anda harus mempelajari sintaks pernyataan MySQL untuk menulis stored procedures.
  - Jika Anda mengulangi logika aplikasi Anda di dua tempat yang berbeda: kode bahasa pemrograman dan kode stored procedures, membuat pemeliharaan aplikasi akan menjadi lebih sulit.
  - Migrasi ke sistem manajemen basis data yang berbeda (DB2, SQL Server, dll) berpotensi menjadi lebih sulit.

#### 4. Trigger

- Pengertian Trigger

Trigger adalah sebuah perintah dalam MySQL yang dapat digunakan untuk memicu suatu kejadian atau interaksi atau perintah lain untuk bekerja dalam database MySQL. Aksi atau kejadian tersebut dapat berupa insert, update, dan delete yang telah di deklarasikan sebelumnya, aksi ini dapat ditempatkan pada saat aksi setelah kejadian pemicu ataupun sebelumnya.

Alasan penggunaan trigger adalah untuk Membudayakan dalam hal insert, update ataupun delete pada tabel berbeda dengan parameter kunci yang sama, membuat integritas constraint yang kompleks, mencatat aktivitas suatu tabel, dan sinkronisasi antar tabel.

- Fungsi Trigger

Trigger dapat ditemui ketika melakukan perintah eksekusi table. Hal itu menjadi lebih bermanfaat trigger menjaikan penulisan pemrograman yang sederhana dan dapat menjaga informasi agar tetap konsisten dalam database. Selain itu, terdapat beberapa fungsi trigger yang akan dijelaskan dibawah ini:

- Integritas Data

Dengan adanya trigger, Anda dapat mempertahankan integritas table yang terdapat dalam database. Sebab, Anda dapat melakukan operasi-operasi yang berkaitan dengan pengolahan database, seperti INSERT, UPDATE dan DELETE

- Mencegah Error

Trigger dalam MySQL dapat mencegah terjadinya error dalam pengoperasian data. Jika terjadi error dalam pendefinisian trigger, error tersebut tidak mengganggu trigger yang sedang berjalan.

- Membuat Tugas Kerja menjadi Terjadwal

Ketika trigger telah dijalankan, maka Anda dapat menggunakan berbagai bahasa pemrograman tanpa harus bingung bagaimana cara mengaksesnya.

- Mencegah Proses Transaksi yang Tidak Sah

Dalam praktiknya, biasa trigger digunakan untuk melakukan proses transaksi. Anda dapat menyimpan record transaksi tersebut ke table lain (history) tanpa harus takut jika data tersebut di-update atau delete. Semua perubahan yang terjadi juga dapat diacak berdasarkan waktu pembuatannya.

## 5. Concurrency

Database MS mengijinkan banyak transaksi pada saat bersamaan untuk mengakses data yang sama. Concurrency Control Mechanism atau CCM dibutuhkan agar transaksi tidak saling bertumbukan atau mengganggu satu sama lainnya.

Tiga masalah dalam Concurrency yaitu :

1. Lost Update Problem
2. Uncommitted dependency problem
3. Inconsistent analysis problem

### Locking

Jika sebuah transaksi ingin record atau resource tidak berubah dalam waktu tertentu maka dia meminta lock. Jenis-jenis locking adalah Exclusive Lock atau Xlock dan Shared lock atau Slock.

## Isolation Level

Isolation level adalah derajat pengaruh antar transaksi. Tertinggi yaitu serializable atau sama sekali tidak mempengaruhi, dan seolah-olah dikerjakan secara beruntun.

Maka solusinya kompromi isolation level.

## Kompromi isolation level

- |                    |  |
|--------------------|--|
| Dirty Read         | : Dapat melihat record yang sudah di rollback transaksi lain atau yang belum di commit.                      |
| Nonrepeatable read | : T1 baca sebuah record, record tersebut diubah transaksi lain, saat T1 membaca lagi nilainya sudah berubah. |
| Phantoms           | : Retrieve menghasilkan record baru yang sebelumnya tidak dikenal atau tidak ada.                            |