MODUL

DATABASE

(CCD211)

MODUL SESI VI

SQL: DATA DEFINITION

DISUSUN OLEH

NOVIANDI, S.Kom, M.Kom

UNIVERSITAS ESA UNGGUL

2020

BAB VI
SQL: Data Definition

Tujuan

1. Tipe data yang didukung oleh standar SQL.
2. Tujuan fitur peningkatan integritas
3. Bagaimana mendefinisikan batasan integritas menggunakan SQL.
   - Required data;
   - Domain constraints;
   - Entity integrity;
   - Referential integrity;
   - General constraints.
4. Cara menggunakan fitur penyempurnaan pernyataan CREATE dan TABLE.
5. Tujuan views.
6. Cara membuat dan menghapus view menggunakan SQL.
7. Bagaimana DBMS melakukan operasi pada view.
8. Dalam kondisi apa view dapat diupdate.
9. Keuntungan dan kerugian views.
10. Bagaimana model transaksi ISO bekerja.
11. Bagaimana menggunakan pernyataan GRANT dan REVOKE sebagai tingkat keamanan.

Teori

DATA TYPE	DECLARATIONS				
boolean	BOOLEAN				
character	CHAR	VARCHAR			
bit†	BIT	BIT VARYING			
exact numeric	NUMERIC	DECIMAL	INTEGER	SMALLINT	BIGINT
approximate numeric	FLOAT	REAL	DOUBLE PRECISION		
datetime	DATE	TIME	TIMESTAMP		
interval	INTERVAL				
large objects	CHARACTER LARGE OBJECT		BINARY LARGE OBJECT		

BIT and BIT VARYING have been removed from the SQL:2003 standard.

Gambar 6.1 Jenis Data ISO SQL

Integrity Enhancement Feature

Consider five types of integrity constraints:

- Required data
- Domain constraints
- Entity integrity
- Referential integrity
- General constraints

Required data

Position varchar(10) NOT NULL

Domain Constraints

a. CHECK

Sex CHAR NOT NULL

CHECK(sex IN('M','F'))

b. CREATE DOMAIN

CREATE DOMAIN DomainName[AS] dataType

[DEFAULT defaultOption]

[CHECK(searchCondition)]

For example

CREATE DOMAIN SexType AS CHAR

CHECK (VALUE IN('M',"F"));

Sex SexType NOTNULL

Search Condition can involve a table looup:

CREATE DOMAIN BranchNo AS CHAR(4)

CHECK (VALUE IN (SELECT branchNo From Branch));

Domains can be removed using DROP DOMAIN:

DROP DOMAIN DomainName

[RESTRICT | CASCADE]

Integrity Enhancement Feature - Entity Integrity

- Primary key of a table must contain a unique, non-null value for each row
- ISO standard supports FOREIGN KEY clause in CREATE and ALTER TABLE statements:

  PRIMARY KEY(staffNo)

  PRIMARY KEY(clientNo, propertyNo)
- Can only have one PRIMARY KEY clause per table. Can still ensure uniqueness for alternate keys using UNIQUE:

  UNIQUE(telNo)

Integrity Enhancement Feature - Referential Integrity

- FK is column or set of columns that links each row in child table containing foreign FK to row of parent table containing matching PK.
- Referential integrity means that, if FK contains a value, that value must refer to existing row in parent table.
- ISO standard supports definition of FKs with FOREIGN KEY clause in CREATE and ALTER TABLE:

  FOREIGN KEY(branchNo) REFRENCES Branch
- Any INSERT/UPDATE attempting to create FK value in child table without matching CK value in parent is rejected.
- Action taken attempting to update/delete a CK value in parent table with matching rows in child is dependent on referential action specified using ON UPDATE and ON DELETE subclauses:
  - CASCADE

    Delete row from parent and delete matching rows in child, and so on in cascading manner.
  - SET DEFAULT

    Delete row from parent and set each component of FK in child to specified default. Only
  - SET NULL

    Delete row from parent and set FK column(s) in child to NULL. Only valid if FK columns are NOT NULL
  - NO ACTION

Reject delete from parent. Default

FOREIGN KEY (staffNo) REFERENCES Staff ON DELETE SET NULL
FOREIGN KEY (ownerNo) REFERENCES Owner ON UPDATE CASCADE

Integrity Enhancement Feature - General Constraints

- Cloud use CHECK/UNIQUE in CREATE and ALTER TABLE
- Similar to the CHECK clause, also have:
  CREATE ASSERTION AssertionName
  CHECK(searchCondition)

  CREATE ASSERTION StaffNotHandlingTooMuch
  CHECK (NOT EXISTS (SELECT staffNo
  FROM PropertyForRent
  GROUP BY staffNo
  HAVING COUNT(*) > 100))

Data Definition

- SQL DDL allows database objects such as schemas, domains, tables, views, and indexes to be created and destroyed.
- Main SQL DDL statements are:
  CREATE SCHEMA DROP SCHEMA
  CREATE/ALTER DOMAIN DROP DOMAIN
  CREATE/ALTER TABLE DROP TABLE
  CREATE VIEW DROP VIEW
- Many DBMSs alsi provide
  DROP INDEX DROP INDEX
- Relations and other database objects exist in an environment
- Each environment contains one or more catalogs, and each catalog consists of set of schemas
- Schema is named collection of related database objects
- Objects in a schema can be tables, views, domains, assertions, collations, translations, and character sets. All have same owner

CREATE SCHEMA

CREATE SCHEMA [Name | AUTHORIZATION Creatorld]

DROP SCHEMA Name [RESTRICT | CASCADE]

- With RESTRICT (default), schema must be empty or operation fails
- With CASCADE, operation cascades to drop all objects associated with schema in order defined above. If any of these operations fail, DROP SCHEMA fails.

CREATE TABLE

CREATE TABLE TableName

{(columnName dataType [**NOT NULL**] **[UNIQUE]**

[DEFAULT defaultOption] **[CHECK** (searchCondition)] [, . . .]}

[PRIMARY KEY (listOfColumns),]

{[UNIQUE (listOfColumns)] [, . . .]}

{[FOREIGN KEY (listOfForeignKeyColumns)

REFERENCES ParentTableName [(listOfCandidateKeyColumns)]

[MATCH {PARTIAL | **FULL**}

[ON UPDATE referentialAction]

[ON DELETE referentialAction]] [, . . .]}

{[CHECK (searchCondition)] [, . . .]})

1. Creates a table with one or more columns of the specified datatype
2. With NOT NULL, system rejects any attempt to insert a null in the column
3. Can specify a DEFAULT value for the column
4. Primary keys should always be specified as NOT NULL
5. FOREIGN KEY clause specifies FK along with the referential action

   Contoh:

   Buat tabel PropertyForRent menggunakan fitur yang tersedia dari pernyataan CREATE TABLE.

```
CREATE DOMAIN OwnerNumber AS VARCHAR(5)
        CHECK (VALUE IN (SELECT ownerNo FROM
        PrivateOwner));
CREATE DOMAIN StaffNumber AS VARCHAR(5)
        CHECK (VALUE IN (SELECT staffNo FROM Staff));
```

```
CREATE DOMAIN BranchNumber AS CHAR(4)
        CHECK (VALUE IN (SELECT branchNo FROM Branch));
CREATE DOMAIN PropertyNumber AS VARCHAR(5);
CREATE DOMAIN Street AS VARCHAR(25);
CREATE DOMAIN City AS VARCHAR(15);
CREATE DOMAIN Postcode AS VARCHAR(8);
CREATE DOMAIN PropertyType AS CHAR(1)
        CHECK(VALUE IN ('B', 'C', 'D', 'E', 'F', 'M', 'S'));
CREATE DOMAIN PropertyRooms AS SMALLINT;
        CHECK(VALUE BETWEEN 1 AND 15);
CREATE DOMAIN PropertyRent AS DECIMAL(6,2)
        CHECK(VALUE BETWEEN 0 AND 9999.99);
CREATE TABLE PropertyForRent(
propertyNo      PropertyNumber      NOT NULL,
street          Street              NOT NULL,
city            City                NOT NULL,
postcode        PostCode,
type            PropertyType        NOT NULL DEFAULT 'F',
rooms           PropertyRooms       NOT NULL DEFAULT 4,
rent            PropertyRent        NOT NULL DEFAULT 600,
ownerNo         OwnerNumber         NOT NULL,
staffNo         StaffNumber
                CONSTRAINT StaffNotHandlingTooMuch
                    CHECK (NOT EXISTS (SELECT staffNo
                                       FROM PropertyForRent
                                       GROUP BY staffNo
                                       HAVING COUNT(*) > 100)),
branchNo        BranchNumber        NOT NULL,
PRIMARY KEY (propertyNo),
FOREIGN KEY (staffNo) REFERENCES Staff ON DELETE SET NULL
                                       ON UPDATE CASCADE,
FOREIGN KEY (ownerNo) REFERENCES PrivateOwner ON DELETE
NO
                                       ACTION ON UPDATE CASCADE,
```

FOREIGN KEY (branchNo) **REFERENCES** Branch **ON DELETE NO ACTION ON UPDATE CASCADE);**

ALTER Tabel

1. Add a new column to a table
2. Drop a column from a table
3. Add a new table constraint
4. Drop a table constraint
5. Set a default for a column
6. Drop a default for a column

Format dasar ALTER

ALTER TABLE TableName
[ADD [**COLUMN**] columnName dataType **[NOT NULL] [UNIQUE]**
[DEFAULT defaultOption**] [CHECK** (searchCondition)]]
[DROP [**COLUMN**] columnName **[RESTRICT** | **CASCADE**]]
[ADD [CONSTRAINT [ConstraintName]] tableConstraintDefinition]
[DROP CONSTRAINT ConstraintName **[RESTRICT** | **CASCADE**]]
[ALTER [**COLUMN**] **SET DEFAULT** defaultOption**]**
[ALTER [**COLUMN**] **DROP DEFAULT]**

DROP COLUMN menentukan nama kolom yang akan dihapus dari definisi tabel, dan memiliki kualifikasi opsional yang menentukan apakah tindakan DROP akan menurun atau tidak:

- RESTRICT: Operasi DROP ditolak jika kolom direferensikan oleh objek database lain (misalnya, oleh definisi tampilan). Ini adalah pengaturan default.
- CASCADE: Operasi DROP melanjutkan dan secara otomatis menjatuhkan kolom dari objek database yang direferensikan. Operasi ini berjenjang, sehingga jika kolom dijatuhkan dari objek referensi, SQL memeriksa apakah kolom tersebut direferensikan oleh objek lain dan melepaskannya dari sana jika ada, dan seterusnya.

Contoh ALTER Tabel

a. Ubah tabel Staf dengan menghapus default 'Asisten' untuk kolom posisi dan menyetel default untuk kolom jenis kelamin menjadi perempuan ('F').

ALTER TABLE Staff

ALTER position **DROP DEFAULT;**

ALTER TABLE Staff

ALTER sex **SET DEFAULT** ‘F’;

b. Ubah tabel PropertyForRent dengan menghapus constraint bahwa staf tidak diperbolehkan untuk menangani lebih dari 100 properti sekaligus. Ubah tabel Klien dengan menambahkan kolom baru yang mewakili jumlah kamar yang diinginkan.

ALTER TABLE PropertyForRent

DROP CONSTRAINT StaffNotHandlingTooMuch;

ALTER TABLE Client

ADD prefNoRooms PropertyRooms;

Pernyataan ALTER TABLE tidak tersedia di semua dialek SQL. Dalam beberapa dialek, pernyataan ALTER TABLE tidak dapat digunakan untuk menghapus kolom yang ada dari tabel. Dalam kasus seperti itu, jika kolom tidak lagi diperlukan, kolom tersebut dapat diabaikan begitu saja tetapi tetap dalam definisi tabel. Namun, jika Anda ingin menghapus kolom dari tabel, Anda harus:

- Upload semua data dari table;
- Remove definisi tabel menggunakan drop table statement;
- Definisikan kembali table yang baru menggunakan statement create table;
- Reload kembali data ke table yang baru.

Removing a Table (DROP TABLE)

DROP TABLE TableName [RESTRICT | CASCADE]

Contoh notasi DROP:

DROP TABLE PropertyForRent;

- Remove named table and all rows within it
- With RESTRICT, if any other objects depend for their existence on continued existence of this table, SQL does not allow request

- With CASCADE, SQL drops all dependent objects (and objects dependent on these objects).

Views

- Hasil dinamis dari satu atau lebih operasi relasional yang beroperasi pada relasi dasar untuk menghasilkan relasi lain.
- View adalah relasi virtual yang tidak harus ada dalam database tetapi dapat diproduksi atas permintaan pengguna tertentu, pada saat diminta.
- Konten views didefinisikan sebagai queri pada satu atau beberapa relasi dasar.
- Dengan resolusi view, setiap operasi pada view secara otomatis diterjemahkan ke dalam operasi pada relasi yang darinya ia berasal.
- Dengan materialisasi view, view tersebut disimpan sebagai tabel sementara, yang dipertahankan saat tabel dasar yang mendasarinya diperbarui.

CREATE VIEW

```
CREATE VIEW ViewName[(newColumnName[,…])]
    AS subselect
    [WITH [CASCADE | LOCAL] CHECK OPTION]
```

- Dapat memberikan nama untuk setiap kolom dalam view
- Jika daftar nama kolom ditentukan, itu harus memiliki jumlah item yang sama dengan jumlah kolom yang dihasilkan oleh subselect
- Jika dihilangkan, setiap kolom mengambil nama kolom yang sesuai di subselect
- Daftar harus ditentukan jika ada ambiguitas dalam nama kolom.
- Subselect ini dikenal sebagai query yang menentukan.
- WITH CHECK OPTION memastikan bahwa jika baris gagal memenuhi klausa WHERE untuk menentukan queri, baris tersebut tidak ditambahkan ke tabel dasar yang mendasarinya.
- Membutuhkan hak istimewa SELECT pada semua tabel yang direferensikan dalam subselect dan hak istimewa USAGE pada domain mana pun yang digunakan dalam kolom yang direferensikan

Contoh Create Horizontal Vies

Buat view sehingga manajer di cabang B003 hanya dapat melihat detail untuk staf yang bekerja di kantor cabangnya.

```
CREATE VIEW Manager3Staff
AS SELECT *
FROM Staff
WHERE branchNo = 'B003';
```

Menampilkan semua data pada table Manager3Staff

```
SELECT * FROM Manager3Staff;
```

Hasil

staffNo	fName	lName	position	sex	DOB	salary	branchNo
SG37	Ann	Beech	Assistant	F	10-Nov-60	12000.00	B003
SG14	David	Ford	Supervisor	M	24-Mar-58	18000.00	B003
SG5	Susan	Brand	Manager	F	3-Jun-40	24000.00	B003

Contoh Menampilkan Vertical View

Buat view detail staf di cabang B003 yang mengecualikan informasi gaji, sehingga hanya manajer yang dapat mengakses detail gaji untuk staf yang bekerja di cabang mereka.

```
CREATE VIEW Staff3
AS SELECT staffNo, fName, lName, position, sex
FROM Staff
WHERE branchNo = 'B003';
```

Hasil:

staffNo	fName	lName	position	sex
SG37	Ann	Beech	Assistant	F
SG14	David	Ford	Supervisor	M
SG5	Susan	Brand	Manager	F

Note

Bahwa kita bisa menulis ulang pernyataan ini untuk menggunakan tampilan Manager3Staff daripada tabel Staff, dengan demikian:

CREATE VIEW Staff3
AS SELECT staffNo, fName, lName, position, sex
FROM Manager3Staff;

Grouped and Joined Views

Buat view staf yang mengelola properti untuk disewakan, termasuk *branch number* tempat mereka bekerja, *staff number*, dan *number of properties* yang mereka kelola.

CREATE VIEW StaffPropCnt (branchNo, staffNo, cnt)
AS SELECT s.branchNo, s.staffNo, **COUNT**(*)
FROM Staff s, PropertyForRent p
WHERE s.staffNo = p.staffNo
GROUP BY s.branchNo, s.staffNo;

Hasil:

branchNo	staffNo	cnt
B003	SG14	1
B003	SG37	2
B005	SL41	1
B007	SA9	1

Removing a View (DROP VIEW)

View dihapus dari database dengan pernyataan DROP VIEW:

DROP VIEW ViewName **[RESTRICT | CASCADE]**

- Cuases definition of view to be deleted from database
  Contoh

DROP VIEW Manager3Staff;

- Dengan CASCADE, semua objek dependen terkait dihapus; yaitu setiap tampilan yang ditentukan pada tampilan yang dijatuhkan
- Dengan RESTRICT (default), jika ada objek lain yang bergantung pada keberadaan mereka pada keberadaan terus tampilan yang dijatuhkan, perintah ditolak.

Keuntungan Views

- Independensi data
- Mata Uang
- Peningkatan keamanan
- Mengurangi kompleksitas
- Kenyamanan
- Kustomisasi
- Integritas data

Kekurangan of Views

- Update batasan
- Pembatasan struktur
- Perfomance

Grant atau Revoke Hak Istimewa

- DCL commands are used to enforce database security in a multiple database environment.
- Two types of DCL commands are
  - Grant
  - Revoke
- Database Administrator's or owner's of the database object can provide/remove privileges on a database object.

Grant

Grant berfungsi untuk membuat user baru dan memberikan hak istimewa. Grant adalah salah satu privilege untuk tabel. Grant digunakan untuk memberikan privilege kepada tabel yang didefinisikan kepada pemakai lain. Privilege untuk pemakai dalam perintah grant didefinisikan dengan menggunakan nama-nama privilege. Nama privilege memudahkan administrator untuk dapat memberikan priivilege tanpa harus tahu apa nama field dan tabel yang harus diisi.

Perintah grant secara otomatis akan menambah data pemakai apabila data nama pemakai yang disertakan pada perintah tersebut belum ada dalam tabel user. Perintah grant memudahkan administrator untuk tidak perlu melakukan perintah pendefinisian privilege dengan menggunakan sql. Karena dengan menggunakan sql, kita harus hafal nama tabel yang harus diisi, field apa saja yang harus diisi, jumlah field yang harus diisi. Kesalahan mudah dilakukan dengan menggunakan perintah sql karena ketidaktelitian atau ketidakhafalan nama tabel dan nama field yang harus diisi.

Sintak Umum:

- GRANT *hak_akses* ON *nama_tabel* TO *pemakai*;
- GRANT ALL PRIVILEGES ON database_name.* TO ‘myuser’ IDENTIFIED BY ‘mypassword’;

Contoh Penggunaan:

1. GRANT SELECT ON Point_Of_Sales.jenis TO Febe;
2. GRANT SELECT ON Point_Of_Sales.jenisTO Winda;
3. GRANT SELECT ON Point_Of_Sales.item TO Elfrida;
4. GRANT ALL PRIVILEGES ON Point_Of_Sales.User TO Admin;
5. GRANT ALL ON Point_Of_Sales.jualDetail TO Admin
6. SHOW GRANTS FOR root@localhost;
7. SHOW GRANTS FOR Admin;
8. GRANT SELECT,INSERT ON Point_Of_Sales.jualDetail TO kasir;
9. GRANT SELECT(Kode,Nama) ON Point_Of_Sales.jenis TO Elfrida;
10. GRANT UPDATE(kodeItem,NmItem,kategori,Harga) ON Point_Of_Sales.item TO Elfrida;

REVOKE

Untuk menghapus batasan hak akses yang telah diatur dengan menggunakan perintah GRANT, digunakan perintah **REVOKE**.

Sintak umum:

- REVOKE *hak_akses* ON *nama_tabel* FROM *……;*

  Atau menghapus batasan hak akses untuk database & tabel :
- REVOKE *hak_akses* ON *nama_database.nama_tabel* FROM *user;*

  Atau menghapus batasan hak akses untuk kolom tertentu :
- REVOKE *hak_akses(field1,field2, field3,…)* ON *nama_database.nama_tabel* FROM *user*;

Penulisan perintah REVOKE:

- **HakAkses(field)** : kita harus memberikan sedikitnya satu hak akses. Untuk setiap hak akses yang diberikan, dapat juga diberikan daftar field yang diletakkan dalam kurung, dan dipisahkan dengan tanda koma.

  Contoh: REVOKE select (nim, nama), update, insert(nim), …
- **NamaTabel :** merupakan nama tabel yang dikenal hak akses tersebut. Harus ada sedikitnya satu nama tabel. Dapat menggunakan symbol asterik (*) untuk mewakili semua tabel pada database aktif. Penulisan namaTabel dapat juga diikuti oleh nama database diikuti nama tabel yang dipisahkan dengan tanda titik. Menggunakan simbol *.* berarti semua database dan semua tabel yang dikenai hak akses tersebut.
- **namaAccount@namaHost :** jika nama account tidak ada, tidak pernah diberikan hak akses dengan perintah GRANT sebelumnya maka akan terjadi error.

Contoh Penggunaan :

1. REVOKE SELECT ON Point_Of_Sales.jenis FROM Febe;
2. REVOKE SELECT ON Point_Of_Sales.jenis FROM Winda;
3. REVOKE SELECT ON Point_Of_Sales.item FROM Elfrida;
4. REVOKE ALL PRIVILEGES ON Point_Of_Sales.user FROM Admin;
5. REVOKE ALL ON Point_Of_Sales.jualDetail FROM Admin;

6. REVOKE SELECT,INSERT ON Point_Of_Sales.jualDetail FROM kasir;
7. REVOKE SELECT(Kode,Nama) ON Point_Of_Sales.jenis FROM Elfrida;
8. REVOKE UPDATE(kodeItem,NmItem,kategori,Harga) ON Point_Of_Sales.item FROM Elfrida;
9. REVOKE INSERT ON Point_Of_Sales.jenis FROM Winda;
10. REVOKE ALL ON Point_Of_Sales.item FROM PUBLIC;

Latihan

1. Apa pernyataan DDL SQL utama?
2. Diskusikan fungsionalitas dan pentingnya Fitur Peningkatan Integritas (IFF).
3. Apa hak istimewa yang biasanya diberikan kepada pengguna database?
4. Diskusikan keuntungan dan kerugian dari view.
5. Diskusikan cara-cara penyelesaian transaksi.
6. Batasan apa yang diperlukan untuk memastikan bahwa sebuah view dapat diperbarui?
7. Apa yang dimaksud dengan view terwujud dan apa keuntungan dari mempertahankan view terwujud daripada menggunakan proses resolusi view?
8. Jelaskan perbedaan antara kontrol akses diskresioner dan wajib. Jenis mekanisme kontrol apa yang didukung SQL?
9. Jelaskan bagaimana mekanisme kontrol akses SQL bekerja.