

# Data Definition Language (DDL)

Disusun Oleh:

Wardani Muhamad S.T., M.T. – [wdm@politekniktelkom.ac.id](mailto:wdm@politekniktelkom.ac.id)

Disajikan Oleh:

Dedy Rahman Wijaya

[dedyrw@tass.telkomuniversity.ac.id](mailto:dedyrw@tass.telkomuniversity.ac.id)

Hanya dipergunakan untuk kepentingan pengajaran di Lingkungan Politeknik Telkom

# Indikator Kompetensi

<b>Materi Bahasan yang Diujikan</b>	<b>Kriteria Penilaian</b>				
	<b>Tidak Lulus (0-5)</b>	<b>Dasar (6-10)</b>	<b>Menengah (11-15)</b>	<b>Cukup Mahir (16-20)</b>	<b>Mahir (21-25)</b>
Pendefinisian Skema	Tidak memenuhi kriteria penilaian kompetensi dasar	Menyebutkan empat tipe data dan kegunaannya	Menjelaskan kegunaan tiap keyword DDL	Membuat tabel dengan menggunakan tipe data yang tepat pada beberapa tabel (80% dari keseluruhan tabel pada diagram relationship) lengkap dengan PK	Memodifikasi tabel lengkap dengan PK sesuai skema relasi dan menggunakan tipe data yang tepat
Refferential Integrity (FK)	Tidak memenuhi kriteria penilaian kompetensi dasar	Menentukan tipe data yang tepat pada field FK sesuai terkait referential skema relasi	Menyebutkan aturan-aturan terkait referential integrity (jenis on delete clause pada FK)	Menjelaskan aturan-aturan yang terkait dengan referential integrity, seperti: restrict, set to null, set to default, cascade dan no action	Membuktikan referential integrity pada tabel yang saling berelasi melalui perintah DML (insert dan delete)

# Kelompok SQL Statement

- ▶ Data Definition Language digunakan untuk mendefinisikan **struktur** tabel pada suatu basisdata:
  - ▶ Pembuatan,
  - ▶ Pengubahan, dan
  - ▶ Penghapusan

SELECT  
INSERT  
UPDATE  
DELETE  
MERGE

Data Manipulation Language (DML)

CREATE  
ALTER  
DROP  
RENAME  
TRUNCATE  
COMMENT

Data Definition Language (DDL)

GRANT  
REVOKE

Data Control Language (DCL)

COMMIT  
ROLLBACK  
SAVEPOINT

Transaction Control

# Database Object

- ▶ Sebuah basisdata pada Oracle dapat terdiri atas beberapa struktur data:
  - ▶ **Table:** penyimpanan data
  - ▶ View: subset data dari satu atau beberapa table
  - ▶ Sequence: men-generate nilai numeric secara otomatis
  - ▶ Index: meningkatkan performansi query
  - ▶ Synonym: memberikan nama alternatif pada objek
- ▶ Struktur table pada Oracle
  - ▶ Table dapat dibuat kapan saja
  - ▶ Tidak perlu menentukan ukuran table
  - ▶ Struktur table dapat dimodifikasi secara online

# Struktur Table

employee

Record/baris data

Diagram illustrating the structure of a table named 'employee' with 20 records. The table has columns: EMPLOYEE\_ID, LAST\_NAME, FIRST\_NAME, SALARY, COMMISSION\_PCT, and DEPARTMENT\_ID.

Annotations with numbers 1 through 6 point to specific features:

- 1** Record/baris data (points to the first record)
- 2** Primary Key Field (points to the EMPLOYEE\_ID column)
- 3** Non key field (points to the COMMISSION\_PCT column)
- 4** Foreign Key Field (points to the DEPARTMENT\_ID column)
- 5** Nilai pada column DEPARTMENT\_ID (points to the DEPARTMENT\_ID column values)
- 6** NULL Value (points to the empty cell in the COMMISSION\_PCT column for record 201)

Table Data:

EMPLOYEE_ID	LAST_NAME	FIRST_NAME	SALARY	COMMISSION_PCT	DEPARTMENT_ID
100	King	Steven	24000		90
101	Kochhar	Neena	17000		90
102	De Haan	Lex	17000		90
103	Humold	Alexander	9000		60
104	Ernst	Bruce	6000		60
107	Lorentz	Diana	4200		60
124	Mourgos	Kevin	6500		60
141	Rajs	Trenna	3500		90
142	Davies	Curtis	3100		90
143	Matos	Randall	2800		90
144	Vargas	Peter	2500		90
149	Zlotkey	Eleni	10500	2	80
174	Abel	Ellen	11000	3	80
176	Taylor	Jonathon	8600	2	80
178	Grant	Kimberely	7000	15	80
200	Whalen	Jennifer	4400		10
201	Hartstein	Michael	13000		20
202	Fay	Pat	6000		20
205	Higgins	Shelley	12000		110
206	Gietz	William	8200		110

# Apa yang Harus Diperhatikan?

- ▶ Ukuran data

EMPLOYEE_ID	LAST_NAME	FIRST_NAME	SALARY	COMMISSION_PCT	DEPARTMENT_ID
100	King	Steven	24000		90
101	Kochhar	Neena	17000		90

- ▶ Apa tipe data yang cocok?
- ▶ Berapa panjang/banyak karakter yang akan disimpan pada sebuah column?
- ▶ Apakah ada constraint tertentu pada column?
  - ▶ NOT NULL value
  - ▶ DEFAULT value
- ▶ Relasi dengan tabel lain
  - ▶ Referential integrity → foreign key
- ▶ Pemberian nama: column, tabel, constraint, dll

# Aturan Penamaan

- ▶ Aturan penamaan table dan column
  - ▶ Dimulai dengan abjad
  - ▶ Memiliki panjang karakter 1-30
  - ▶ Terdiri dari A-Z, a-z, 0-9, \_, \$, and #
  - ▶ Tidak sama (duplicate) dengan nama object yang dimiliki oleh user yang sama
  - ▶ Bukan merupakan nama cadangan yang dimiliki oleh Oracle
- ▶ Sifat penamaan adalah case-insensitive.
  - ▶ Sebagai contoh, pemberian nama EMPLOYEES akan sama dengan eMPloyees atau eMpLOYEES.

# Data Type

VARCHAR2(size)	Variable-length character data (A maximum size <i>must</i> be specified: minimum size is 1; maximum size is 4,000.)
CHAR [(size)]	Fixed-length character data of length size bytes (Default and minimum size is 1; maximum size is 2,000.)
NUMBER [(p,s)]	Number having precision <i>p</i> and scale <i>s</i> (The precision is the total number of decimal digits, and the scale is the number of digits to the right of the decimal point; the precision can range from 1 to 38, and the scale can range from -84 to 127.)
DATE	Date and time values to the nearest second between January 1, 4712 B.C., and December 31, 9999 A.D.
LONG	Variable-length character data (up to 2 GB)
CLOB	Character data (up to 4 GB)
RAW(size)	Raw binary data of length size (A maximum size <i>must</i> be specified: maximum size is 2,000.)
LONG RAW	Raw binary data of variable length (up to 2 GB)
BLOB	Binary data (up to 4 GB)
BFILE	Binary data stored in an external file (up to 4 GB)
ROWID	A base-64 number system representing the unique address of a row in its table

# Create Table Statement

```
CREATE TABLE [schema.]table  
(column datatype [DEFAULT expr][, ...]);
```

**Schema** Sama dengan nama pemilik

**Table** Nama table

**DEFAULT expr** Menentukan nilai default pada field pada saat penambahan (INSERT) data

**Column** Nama column/field

**Datatype** Tipe data column dan panjang



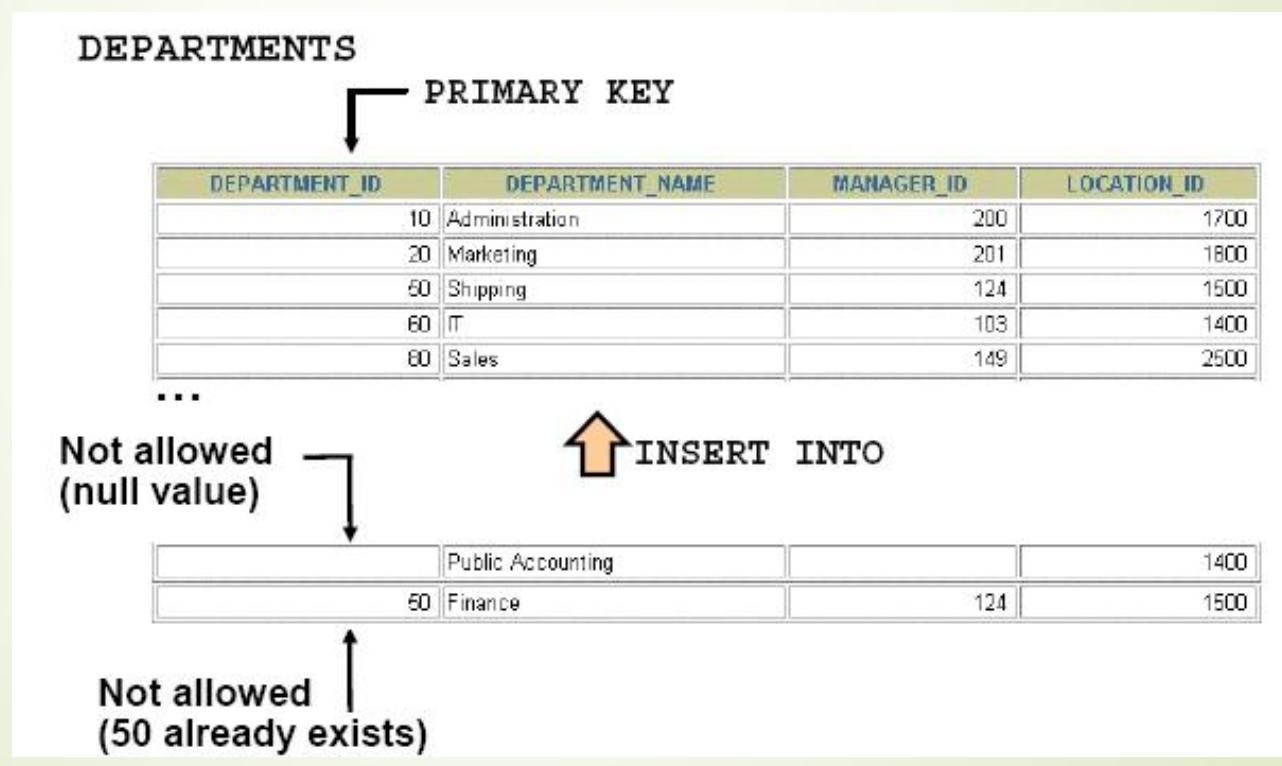
## Table Constraints

- ▶ Table constraint merupakan aturan/ rule yang ada pada suatu tabel
- ▶ Berfungsi untuk menjaga validitas dan integritas data
- ▶ Table constraint dapat ditambahkan pada saat tabel dibuat ataupun setelah tabel dibuat
- ▶ Jenis-jenis table constraints sbb:
  - ▶ PRIMARY KEY
  - ▶ FOREIGN KEY
  - ▶ NOT NULL
  - ▶ UNIQUE
  - ▶ CHECK
  - ▶ DEFAULT

# Primary Key

- ▶ Primary Key merupakan sebuah constraint yang digunakan untuk menjaga keunikan data
- ▶ Data yang tersimpan pada column yang ditunjuk sebagai PK tidak boleh terjadi redundansi
- ▶ Constraint memiliki nama
- ▶ Cara membuat tabel dengan PK:
  - 1 CREATE TABLE mahasiswa1(  
    nim char(4) **PRIMARY KEY**,  
    nama varchar2(30));
  - 2 CREATE TABLE mahasiswa2(  
    nim char(4),  
    nama varchar2(30),  
    **PRIMARY KEY** (nim));
  - 3 CREATE TABLE mahasiswa3(  
    nim char(4),  
    nama varchar2(30),  
    **CONSTRAINT** pk\_mahasiswa **PRIMARY KEY**(nim));

# Skenario Primary Key



# Foreign Key

- ▶ FOREIGN KEY didefinisikan pada column di tabel anak dan merujuk kepada column yang menjadi PK pada tabel induknya.
- ▶ Berfungsi untuk menjaga integritas data dengan menjamin hubungan data antar tabel (*Referential Integrity*)
- ▶ Kondisi yang dapat ditambahkan pada FK:
  - ▶ ON DELETE CASCADE: menghapus data yang bersesuaian pada tabel anak jika data pada tabel induk dihapus
  - ▶ ON DELETE SET NULL: mengubah nilai field (FK) yang bersesuaian menjadi NULL jika data pada tabel induk dihapus
  - ▶ ON DELETE RESTRICT : default nilai field (FK) jika tidak di spesifikasikan. Jika tabel induk punya record di tabel anak, maka tidak bisa di hapus sebelum record di tabel anaknya di hapus.

### Mode CASCADE

► Cara membuat tabel dengan FK:

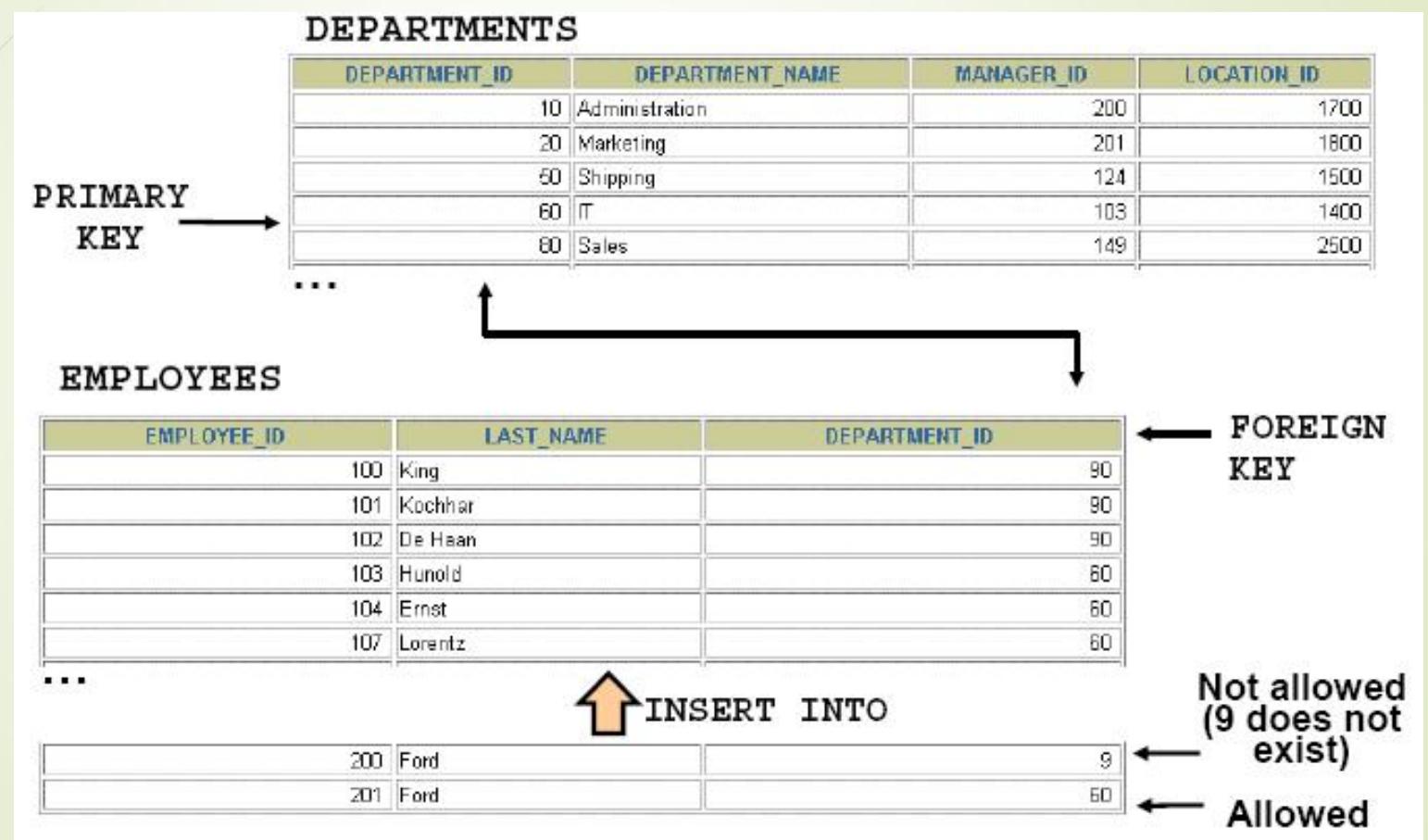
- 1 CREATE TABLE memiliki\_sepeda(  
 id\_sepeda number,  
 nim char(4),  
**FOREIGN KEY(nim) REFERENCES mahasiswa(nim));**
- 2 CREATE TABLE memiliki\_sepeda(  
 id\_sepeda number,  
 nim char(4),  
 CONSTRAINT fk\_memiliki\_mahasiswa  
 FOREIGN KEY (nim) REFERENCES mahasiswa(nim) **ON DELETE CASCADE;**
- 3 CREATE TABLE memiliki\_sepeda(  
 id\_sepeda number,  
 nim char(4),  
 CONSTRAINT fk\_memiliki\_mahasiswa  
 FOREIGN KEY (nim) REFERENCES mahasiswa(nim) **ON DELETE SET NULL;**

Default mode  
 constraint adalah  
**RESTRICT**

MAHASISWA		
PK	nim	nama

MEMILIKI_SEPEDA		
PK	FK	id_sepeda nim

# Skenario Foreign Key



# DEFAULT Option

- ▶ Menentukan nilai default pada kolom pada saat operasi penambahan data.
- ▶ Nilai yang diperbolehkan: Literal values, expressions, atau fungsi SQL.
- ▶ Nilai yang tidak diperbolehkan: nama column yang lain atau pseudocolumn.
- ▶ Tipe data nilai default harus sama dengan tipe data column/field

```
CREATE TABLE hire_dates
  (id NUMBER(8),
  hire_date DATE DEFAULT SYSDATE);
```

```
SQL> CREATE TABLE hire_dates
  2   (id NUMBER(8),
  3    hire_date DATE DEFAULT SYSDATE);
Table created.

SQL> INSERT INTO hire_dates(id) VALUES(1);
1 row created.

SQL> SELECT * FROM hire_dates;
      ID HIRE_DATE
----- 1 09-SEP-13
```

# CHECK Option

- ▶ Menentukan domain nilai yang harus di penuhi oleh sebuah kolom.
- ▶ Nilai yang tidak diperbolehkan: referensi ke curval, nextval, level, rownum, fungsi sysdate, uid, user, userenv serta query ke nilai kolom / baris yang lain.
- ▶ Tipe data nilai default harus sama dengan tipe data column/field

```

CREATE TABLE employees
  ( employee_id      NUMBER(6)
    CONSTRAINT emp_employee_id PRIMARY KEY
  , first_name        VARCHAR2(20)
  , last_name         VARCHAR2(25)
    CONSTRAINT emp_last_name_nn NOT NULL
  , email             VARCHAR2(25)
    CONSTRAINT emp_email_nn    NOT NULL
    CONSTRAINT emp_email_uk    UNIQUE
  , phone_number      VARCHAR2(20)
  , hire_date         DATE
    CONSTRAINT emp_hire_date_nn NOT NULL
  , job_id            VARCHAR2(10)
    CONSTRAINT emp_job_nn      NOT NULL
  , salary             NUMBER(8,2)
    CONSTRAINT emp_salary_ck    CHECK (salary>0)
  )
  
```

# CHECK Option

- ▶ Rule untuk check option bisa lebih dari 1 rule
- ▶ Cth
  - ▶ Tambahkan rule bahwa nilai jenis\_kelamin yang boleh di masukkan hanya L / P dan harus 1 karakter

```
ALTER TABLE mhs ADD CONSTRAINT cek_jenis_kelamin
CHECK (jenis_kelamin IN ('L','P') AND LENGTH(jenis_kelamin)=1)
```

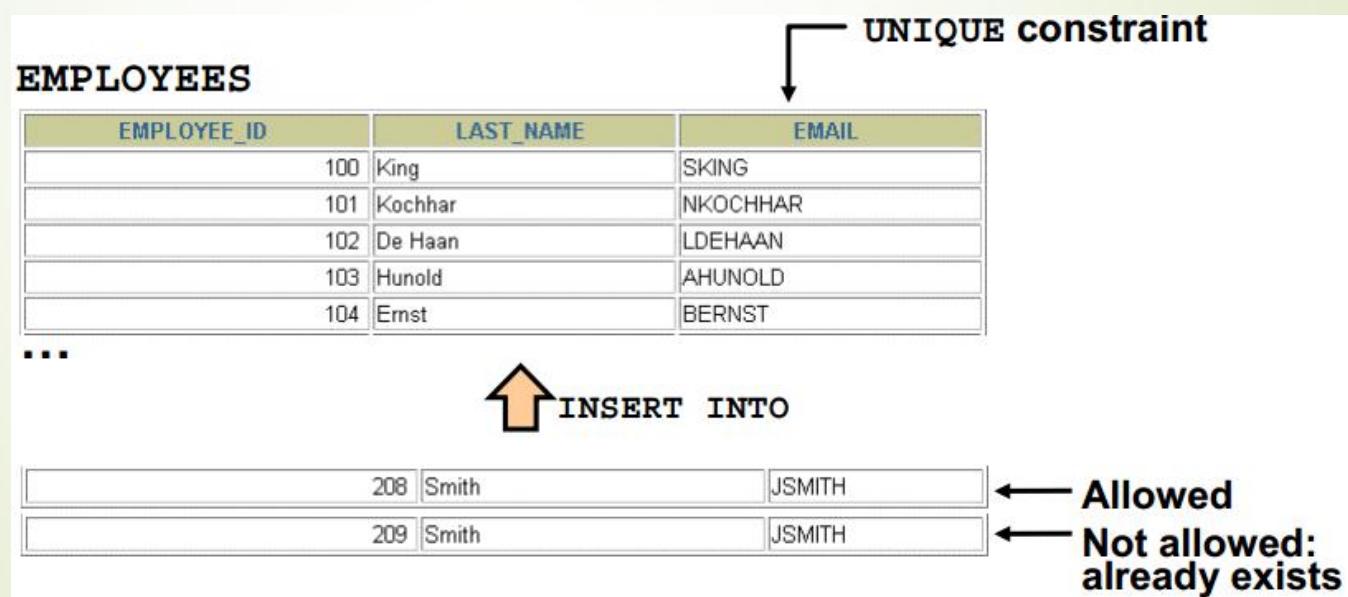
Name	Null?	Type
NIM		VARCHAR2(8)
JENIS_KELAMIN		VARCHAR2(1)

```
SQL> DESC mhs
SQL> INSERT INTO mhs VALUES('30110006','w');
INSERT INTO mhs VALUES('30110006','w')
*
ERROR at line 1:
ORA-02290: check constraint (OP.CEK_JENIS_KELAMIN) violated

SQL> INSERT INTO mhs VALUES('30110006','L');
1 row created.
```

# Unique Constraint

- Constraint yang membatasi bahwa nilai yang di masukkan tidak boleh sama dengan nilai yang sudah ada di tabel sebelumnya untuk kolom tertentu.



# Unique Constraint

**Defined at either the table level or the column level:**

```
CREATE TABLE employees(
    employee_id      NUMBER(6),
    last_name        VARCHAR2(25) NOT NULL,
    email            VARCHAR2(25),
    salary           NUMBER(8,2),
    commission_pct   NUMBER(2,2),
    hire_date        DATE NOT NULL,
    ...
    CONSTRAINT emp_email_uk UNIQUE(email));
```

# NOT NULL Constraint

**Ensures that null values are not permitted for the column:**

EMPLOYEE_ID	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	DEPARTMENT_ID
100	King	SKING	515.123.4567	17-JUN-87	AD_PRES	24000	90
101	Kochhar	NKOCHHAR	515.123.4568	21-SEP-89	AD_VP	17000	90
102	De Haan	LDEHAAN	515.123.4569	13-JAN-93	AD_VP	17000	90
103	Hunold	AHUNOLD	590.423.4567	03-JAN-90	IT_PROG	9000	60
104	Ernst	BERNST	590.423.4568	21-MAY-91	IT_PROG	6000	60
178	Grant	KGRANT	011.44.1644.429263	24-MAY-99	SA_REP	7000	
200	Whalen	JWHALEN	515.123.4444	17-SEP-87	AD_ASST	4400	10
...							

20 rows selected.



**NOT NULL constraint  
(No row can contain  
a null value for  
this column.)**



**NOT NULL  
constraint**



**Absence of NOT NULL  
constraint  
(Any row can contain  
a null value for this  
column.)**

# Mengubah Struktur Table

- ▶ Untuk mengubah struktur table, digunakan perintah ALTER
- ▶ Pengubahan hanya dapat dilakukan pada table yang sudah ada
- ▶ Pengubahan yang dapat dilakukan:
  - ▶ Menambahkan column baru
  - ▶ Memodifikasi column yang sudah ada (mengubah nama column, mengubah tipe data)
  - ▶ Mendefinisikan nilai default pada column baru
  - ▶ Menghapus column
  - ▶ Menambahkan constraint

## Mengubah Struktur Table (2)

- ▶ Menambahkan column baru
  - ▶ Column baru ditambahkan pada urutan terakhir
  - ▶ Harus menentukan tipe data pada column yang ditambahkan
  - ▶ SQL Statement:

```
ALTER TABLE [schema.]table
ADD (column datatype [DEFAULT expr][, ...]);
```
- ▶ Mengubah nama column
  - ▶ Pengubahan nama tidak mengubah tipe data yang telah didefinisikan
  - ▶ SQL Statement:

```
ALTER TABLE [schema.]table
RENAME [old column name] TO [new column name];
```

# Mengubah Struktur Table (3)

## ▶ Menghapus column

- ▶ Jumlah column pada sebuah table minimal satu
- ▶ Tidak dapat dikembalikan (ROLL BACK)
- ▶ SQL Statement:

```
ALTER TABLE [schema.]table  
    DROP ([column-1],[column-2],...,[colum-n]);
```

## ▶ Mengubah tipe data pada suatu column

- ▶ Nama column tidak berubah
- ▶ Pengubahan tipe data harus memperhatikan constraint (PK,FK) yang melekat pada column
- ▶ SQL Statement:

```
ALTER TABLE [schema.]table  
    MODIFY (column datatype [DEFAULT expr]);
```

# Mengubah Struktur Table (3)

- ▶ Menambahkan / Menghapus constraint
  - ▶ Menambah constraint
  - ▶ Cth

```
SQL> ALTER TABLE hire_dates ADD CONSTRAINT pk_hire_date PRIMARY KEY (id);  
Table altered.
```

**Nama  
constraint**

- ▶ Menghapus constraint

```
SQL> ALTER TABLE hire_dates DROP CONSTRAINT pk_hire_date ;  
Table altered.
```

**Nama  
constraint**

# Mengosongkan Data pada Tabel

- ▶ SQL Statement:

```
TRUNCATE TABLE [schema.]table
```

- ▶ Digunakan untuk menghapus keseluruhan baris data (row) pada table
- ▶ Data yang dihapus tidak dapat dikembalikan (ROLL BACK)
- ▶ Tidak mengubah struktur table

# Menghapus Table

- ▶ Menghapus seluruh data dan struktur pada sebuah table
- ▶ Seluruh transaksi yang tertunda akan di-commit
- ▶ Seluruh index dihapus
- ▶ Seluruh constraint dihapus
- ▶ Tidak dapat dikembalikan (ROLL BACK)
- ▶ Perhatikan referential integrity pada table yang dihapus

```
DROP TABLE [schema.]table
```

# Memanfaatkan metadata **USER\_CONSTRAINTS**

- ▶ Untuk melihat constraints apa yang sudah di buat untuk semua objek yang berada di skema user tersebut.
- ▶ **SELECT \* FROM user\_constraints;**

OWNER	CONSTRAINT_NAME	CONSTRAINT_TYPE	TABLE_NAME	SEARCH_CONDITION	R_OWNER	R_CONSTRAINT_NAME	DELETE_RULE	STATUS	DEFERRABLE	DEFERRED	VALIDATED	GENERATED
OP	SYS_C007195	C	REGIONS	"REGIONID" IS NOT NULL				ENABLED	NOT DEFERRABLE	IMMEDIATE	VALIDATED	GENERATED NAME
OP	SYS_C007196	C	REGIONS	"REGIONDESC" IS NOT NULL				ENABLED	NOT DEFERRABLE	IMMEDIATE	VALIDATED	GENERATED NAME
OP	PK_REGIONS	P	REGIONS					ENABLED	NOT DEFERRABLE	IMMEDIATE	VALIDATED	USER NAME
OP	SYS_C007198	C	TERRITORIES	"TERRITORYID" IS NOT NULL				ENABLED	NOT DEFERRABLE	IMMEDIATE	VALIDATED	GENERATED NAME
OP	SYS_C007199	C	TERRITORIES	"TERRITORYDESC" IS NOT NULL				ENABLED	NOT DEFERRABLE	IMMEDIATE	VALIDATED	GENERATED NAME
OP	SYS_C007200	C	TERRITORIES	"REGIONID" IS NOT NULL				ENABLED	NOT DEFERRABLE	IMMEDIATE	VALIDATED	GENERATED NAME
OP	PK_TERRITORIES	P	TERRITORIES					ENABLED	NOT DEFERRABLE	IMMEDIATE	VALIDATED	USER NAME
OP	FK_REGION_TERRID	R	TERRITORIES		OP	PK_REGIONS	NO ACTION	ENABLED	NOT DEFERRABLE	IMMEDIATE	VALIDATED	USER NAME

# Referensi/Daftar Pustaka

- ▶ Oracle Database 10g. SQL Fundamental 1 – Student Guide Volume 1. California: Oracle Publisher 2004.
- ▶ Oracle Database 10g. SQL Reference 10g Release 2. California: Oracle Publisher 2005.