Perubahan Data

satriyo@live.undip.ac.id

Evaluasi Materi

- Data Definition Language (DDL)
  - Create Table
  - Alter Table (Add, Modify, Drop --- Kolom, Constrain, Default)
  - Drop Table
- Data Manipulation Language (DML)
  - Select, From, Where (Projection, Selection, Distinct, *, Alias)
  - Pengurutan Order By
  - Insert Data(Insert into, script import, export)
  - Operator Search (=<>!, between and, in, not in, like %_, escape)
- On Going
  - Perubahan Data (Update, Delete)
  - Aggregate Function (count, sum, avg, min, max
  - Group By, Having
  - Union, Intersection
  - Sub Query
  - Multiple Tabel

Perubahan Data

- Update Data kita gunakan untuk melakukan perubahan/manipulasi atas **satu atau banyak data** yang kita gunakan.
- Delete Data kita gunakan untuk menghapus **satu atau banyak data** yang kita miliki

Update Data

- Pernyataan UPDATE digunakan untuk memodifikasi satu atau banyak baris yang ada dalam sebuah tabel
- UPDATE membutuhkan empat nilai:
  - nama tabel
  - nama kolom yang nilainya akan diubah
  - nilai baru untuk setiap kolom yang sedang dimodifikasi
  - suatu ketentuan yang mengidentifikasi baris mana dalam tabel yang akan dimodifikasi
- Nilai baru untuk kolom bisa merupakan hasil dari subkueri satu baris

Update Data

- Contoh yang ditampilkan menggunakan pernyataan UPDATE untuk mengubah nomor telepon dari satu karyawan di tabel karyawan
- Perhatikan bahwa tabel copy_employees digunakan dalam transaksi ini

```
UPDATE copy_employees
SET phone_number= '123456'
WHERE employee_id= 303;
```

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	PHONE_NUMBER
303	Angelina	Wright	123456

Update Data

- Kita dapat mengubah beberapa kolom dan/atau beberapa baris dalam satu pernyataan UPDATE
- Contoh ini mengubah nomor telepon dan nama belakang untuk dua karyawan

```
UPDATE copy_employees
SET phone_number= '654321', last_name= 'Jones'
WHERE employee_id>= 303;
```

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	PHONE_NUMBER
303	Angelina	Jones	654321
304	Test	Jones	654321

Update Data

- Hati-hati saat memperbarui nilai kolom
- **Jika klausa WHERE dihilangkan, setiap baris dalam tabel akan diperbarui**
- Ini mungkin bukan yang dimaksudkan

```
UPDATE copy_employees
SET phone_number= '654321', last_name= 'Jones'
```

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	PHONE_NUMBER
100	Steven	Jones	654321
101	Neena	Jones	654321
102	Lex	Jones	654321
200	Jennifer	Jones	654321
205	Shelley	Jones	654321
206	William	Jones	654321
149	Eleni	Jones	654321
174	Ellen	Jones	654321
...	...	...	...

Memperbarui Kolom dengan nilai dari Subkueri

- Kita dapat menggunakan hasil dari subkueri satu baris untuk memberikan nilai baru untuk kolom yang diperbarui

```
UPDATE copy_employees
SET salary = (SELECT salary FROM copy_employees WHERE employee_id= 100)
WHERE employee_id= 101;
```

- Contoh ini mengubah gaji satu karyawan (id = 101) menjadi gaji yang sama dengan karyawan lain (id = 100)

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	SALARY
100	Steven	King	24000
101	Neena	Kochhar	24000

Memperbarui Dua Kolom dengan Dua Pernyataan Subkueri

- Untuk memperbarui beberapa kolom dalam satu pernyataan UPDATE, dimungkinkan untuk menulis beberapa sub-baris satu baris, satu untuk setiap kolom
- Dalam contoh berikut, pernyataan UPDATE mengubah gaji dan id pekerjaan dari satu karyawan (id = 206) dengan nilai yang sama dengan karyawan lain (id = 205)

Memperbarui Dua Kolom dengan Dua Pernyataan Subkueri

```
UPDATE copy_employees
SET salary = (SELECT salary FROM copy_employees WHERE employee_id= 205),
    job_id= (SELECT job_id FROM copy_employees WHERE employee_id= 205)
WHERE employee_id= 206;
```

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	SALARY	JOB_ID
205	Shelley	Higgins	12000	AC_MGR
206	William	Gietz	12000	AC_MGR

Memperbarui Baris Menggunakan Subkueri Berkorelasi

- Subkueri bisa berdiri sendiri atau berkorelasi
- Dalam subkueri berkorelasi, Anda memperbarui baris dalam tabel berdasarkan pilihan dari tabel yang sama

Memperbarui Baris Menggunakan Subkueri Berkorelasi

- Dalam contoh di bawah ini, salinan kolom nama departemen dibuat di tabel karyawan
- Kemudian data dari tabel departemen asli diambil, disalin, dan digunakan untuk mengisi salinan kolom di tabel karyawan

```
ALTER TABLE copy_employees
ADD (department_namevarchar2(30) NOT NULL);


UPDATE copy_employees e
SET e.department_name = (SELECT d.department_name FROM
departments d WHERE e.department_id= d.department_id);
```

DELETE

- Pernyataan DELETE digunakan untuk menghapus satu atau banyak baris yang ada dalam sebuah tabel
- Pernyataan tersebut membutuhkan dua nilai:
  - nama tabel
  - Ketentuan yang mengidentifikasi baris untuk dihapus

DELETE

- Contoh yang ditampilkan menggunakan tabel salinan karyawan untuk menghapus satu baris —karyawan dengan nomor ID 303

```
DELETE from copy_employees
WHERE employee_id= 303;
```

- Apa yang Anda prediksi akan dihapus jika klausa WHERE dihilangkan dalam pernyataan DELETE?
- **Semua baris dalam tabel dihapus jika Anda menghilangkan klausa WHERE**

Subkueri DELETE

- Subkueri juga dapat digunakan pernyataan DELETE
- Contoh yang ditampilkan menghapus baris dari tabel karyawan untuk semua karyawan yang bekerja di departemen Shipping
- Mungkin departemen ini telah diganti nama atau ditutup

```
DELETE FROM copy_employees
WHERE department_id = (SELECT department_id FROM departments WHERE
department_name= 'Shipping');
```

Subkueri DELETE

- Contoh di bawah ini menghapus baris semua karyawan yang bekerja untuk manajer yang mengelola kurang dari 2 karyawan

```
DELETE FROM copy_employees e
WHERE e.manager_id IN (SELECT d.manager_id
                       FROM employees d
                       HAVING count (d.department_id) < 2
                       GROUP BY d.manager_id);
```

Kesalahan Constraint Integritas

- Constraint integritas memastikan bahwa data sesuai dengan seperangkat aturan yang diperlukan
- Kendala secara otomatis diperiksa setiap kali pernyataan DML yang dapat melanggar aturan dieksekusi
- Jika ada aturan yang akan dilanggar, tabel tidak diperbarui dan kesalahan dikembalikan

Kesalahan Constraint Integritas

- Contoh ini melanggar constraint NOT NULL karena last_name memiliki constraint not null dan id = 999 tidak ada, sehingga subkueri mengembalikan hasil null

```
UPDATE copy_employees
SET last_name= (SELECT last_name
                FROM copy_employees
                WHERE employee_id= 999)
WHERE employee_id= 101;
```

```
ORA-01407: cannot update
("US_A009EMEA815_PLSQL_T01"."COPY_EMPLOYEES"."LAST_NAME") to NULL
```

Kesalahan Constraint Integritas

- Kapan constraint primary key -foreign key diperiksa?
- Tabel EMPLOYEES memiliki constraint foreign key pada department_id yang mereferensikan department_id dari tabel DEPARTMENTS
- Ini memastikan bahwa setiap karyawan milik departemen yang valid
- Dalam tabel DEPARTMENTS, ada department_id 10 dan 20 tetapi 15 tidak

Kesalahan Constraint Integritas

- Manakah dari pernyataan berikut yang mengembalikan kesalahan?

```
1.UPDATE employees SET department_id= 15
WHERE employee_id= 100;

2.DELETE FROM departments WHERE department_id= 10;

3.UPDATE employees SET department_id= 10
WHERE department_id= 20;
```

Kesalahan Constraint Integritas

- Saat memodifikasi salinan tabel Anda (misalnya copy_employees), Anda mungkin tidak melihat kesalahan constraint null, tetapi Anda tidak akan melihat kesalahan primary key -foreign key
- Alasan untuk ini adalah bahwa CREATE TABLE…. Pernyataan AS (SELECT …) yang digunakan untuk membuat salinan tabel, menyalin baris dan bukan constraint null, tetapi tidak menyalin constraint primary key-foreign key
- Oleh karena itu, tidak ada constraint primary key-foreign key di salah satu tabel yang disalin
- Menambahkan constraint dibahas dalam pelajaran lainnya

Klausa FOR UPDATE di Pernyataan SELECT

- Ketika pernyataan SELECT dikeluarkan terhadap tabel database, tidak ada kunci yang dikeluarkan dalam database pada baris yang dikembalikan oleh kueri yang Anda terbitkan
- Sebagian besar waktu ini adalah bagaimana Anda ingin database berperilaku -untuk menjaga jumlah kunci yang dikeluarkan seminimal mungkin
- Namun, kadang-kadang, Anda ingin memastikan tidak ada orang lain yang dapat memperbarui atau menghapus catatan permintaan Anda saat Anda mengerjakan catatan itu

Klausa FOR UPDATE di Pernyataan SELECT

- Ini adalah saat klausa FOR UPDATE digunakan
- Segera setelah kueri Anda dieksekusi, database akan secara otomatis mengeluarkan kunci level-baris eksklusif pada semua baris yang dihasilkan oleh pernyataan SELECT Anda, yang akan ditahan hingga Anda mengeluarkan perintah COMMIT atau ROLLBACK
- Pengingat: Instance online/yang di-host dari APEX akan mengonfirmasi secara otomatis, dan kunci level baris tidak akan muncul

Klausa FOR UPDATE di Pernyataan SELECT

- Jika Anda menggunakan klausa FOR UPDATE di Pernyataan SELECT dengan beberapa tabel di dalamnya, semua baris dari semua baris akan dikunci

```
SELECT e.employee_id, e.salary, d.department_name
FROM employees e JOIN departments d USING
   (department_id)
WHERE job_id= 'ST_CLERK' AND location_id= 1500
   FOR UPDATE
ORDER BY e.employee_id;
```

Klausa FOR UPDATE di Pernyataan SELECT

- Keempat baris ini sekarang dikunci oleh pengguna yang menjalankan pernyataan SELECT sampai pengguna mengeluarkan COMMIT atau ROLLBACK

EMPLOYEE_ID	SALARY	DEPARTMENT_NAME
141	3500	Shipping
142	3100	Shipping
143	2600	Shipping
144	2500	Shipping

Evaluasi Materi

- Data Definition Language (DDL)
  - Create Table
  - Alter Table (Add, Modify, Drop --- Kolom, Constrain, Default)
  - Drop Table
- Data Manipulation Language (DML)
  - Select, From, Where (Projection, Selection, Distinct, *, Alias)
  - Pengurutan Order By
  - Insert Data(Insert into, script import, export)
  - Operator Search (=<>!, between and, in, not in, like %_, escape)
  - **Perubahan Data (Update, Delete)**
- On Going
  - Aggregate Function (count, sum, avg, min, max
  - Group By, Having
  - Union, Intersection
  - Sub Query
  - Multiple Tabel