

# 1

**Me-Retrieve Data Menggunakan  
Pernyataan SQL SELECT**

ORACLE

Copyright © 2004, Oracle. All rights reserved.

## Tujuan

Setelah menyelesaikan pelajaran ini, Anda akan bisa melakukan sebagai berikut :

- Mendaftar kemampuan dari pernyataan-pernyataan SQL `SELECT`
- Menjalankan pernyataan `SELECT` dasar
- Membedakan antara pernyataan-pernyataan SQL dan perintah-perintah *iSQL\*Plus*

ORACLE

Copyright © 2004, Oracle. All rights reserved.

## Tujuan

Untuk mengeluarkan data dari database, Anda perlu untuk menggunakan pernyataan `SELECT` *Structure Query Language* (SQL). Anda mungkin perlu membatasi kolom yang akan Anda tampilkan. Pada pelajaran ini dijelaskan semua tentang pernyataan SQL yang dibutuhkan untuk melakukan tindakan-tindakan tersebut. Anda mungkin ingin untuk membuat pernyataan `SELECT` yang dapat digunakan lebih dari satu kali.

Pelajaran kali ini juga mencakup hal tentang *iSQL\*Plus environment* (lingkungan) dimana Anda mengeksekusi pernyataan-pernyataan SQL.

## Kemampuan-Kemampuan dari Pernyataan SQL SELECT

### Projection


Table 1

### Selection


Table 1


Table 1

Join


Table 2

ORACLE

Copyright © 2004, Oracle. All rights reserved.

## Kemampuan-Kemampuan dari Pernyataan SQL SELECT

Pernyataan SELECT mengambil (*retrieve*) informasi dari database. Dengan suatu pernyataan SELECT, Anda dapat menggunakan kemampuan berikut :

- **Projection** : Memilih kolom-kolom dalam suatu tabel yang dihasilkan dari suatu query. Memilih beberapa atau banyak kolom yang dibutuhkan.
- **Selection** : Memilih baris-baris dalam suatu tabel yang dihasilkan dari suatu query. Banyak kriteria yang dapat digunakan untuk membatasi baris-baris yang akan diambil.
- **Joining** : Mengambil data bersama yang disimpan dalam tabel-tabel yang berbeda dengan menentukan hubungan antara mereka. *Join-join* SQL dijelaskan lebih detail pada pelajaran selanjutnya.

## Pernyataan SELECT Dasar

```
SELECT *|{ [DISTINCT] column|expression [alias],...}  
FROM table;
```

- **SELECT** mengidentifikasi kolom-kolom yang akan ditampilkan
- **FROM** mengidentifikasi isi tabel di dalam kolom-kolom

ORACLE

Copyright © 2004, Oracle. All rights reserved.

### Pernyataan SELECT Dasar

Dalam bentuk yang mudah, suatu pernyataan SELECT harus menyertakan berikut :

- Suatu klausa SELECT, yang menentukan kolom-kolom yang akan ditampilkan
- Suatu klausa FROM, yang menunjukkan isi kolom pada dalam suatu tabel yang terdaftar dalam klausa SELECT

Dalam sintak :

SELECT	adalah suatu daftar satu atau beberapa kolom
*	memilih semua kolom
DISTINCT	menghilangkan duplikasi
<i>column /expression</i>	memilih nama kolom atau sebuah ekspresi
<i>alias</i>	memberikan nama kolom yang dipilih dengan judul berbeda
FROM <i>table</i>	menentukan isi kolom dalam suatu tabel

**Catatan** : melalui pelajaran ini, kata-kata *keyword*, *clause*, dan *statement* digunakan sebagai berikut :

- Sebuah *keyword* menunjukkan pada elemen individual SQL.  
Contoh, SELECT dan FROM adalah *keywords*.
- Sebuah *clause* adalah bagian dari pernyataan SQL.  
Contoh, SELECT *employee\_id, last\_name, . . .* Adalah suatu klausa.
- Sebuah *Statement* (pernyataan) adalah suatu kombinasi dari dua atau beberapa klausa.  
Contoh, SELECT \* FROM *employees* adalah suatu pernyataan SQL.

## Memilih Semua Kolom

```
SELECT *  
FROM departments;
```

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
10	Administration	200	1700
20	Marketing	201	1800
50	Shipping	124	1500
60	IT	103	1400
80	Sales	149	2500
90	Executive	100	1700
110	Accounting	205	1700
190	Contracting		1700

8 rows selected.

ORACLE

Copyright © 2004, Oracle. All rights reserved.

## Memilih Semua Kolom dari Semua Baris

Anda dapat menampilkan semua kolom-kolom data dalam suatu tabel dengan menggunakan *keyword* `SELECT` diikuti suatu *asterisk* (\*). Contoh pada slide, tabel department berisi empat kolom : `DEPARTMENT_ID`, `DEPARTMENT_NAME`, `MANAGER_ID`, dan `LOCATION_ID`. Tabel berisi tujuh baris, satu untuk setiap departemen.

Anda juga dapat menampilkan semua kolom dalam tabel dengan mendaftar semua kolom setelah *keyword* `SELECT`. Sebagai contoh, pernyataan SQL berikut (seperti contoh pada slide) menampilkan semua kolom dan semua baris dari tabel `DEPARTMENTS` :

```
SELECT department_id, department_name, manager_id, location_id  
FROM departments;
```

## Memilih Kolom-Kolom Tertentu

```
SELECT department_id, location_id  
FROM departments;
```

DEPARTMENT_ID	LOCATION_ID
10	1700
20	1800
50	1500
60	1400
80	2500
90	1700
110	1700
190	1700

8 rows selected.

ORACLE

Copyright © 2004, Oracle. All rights reserved.

## Memilih Kolom-Kolom Tertentu dari Semua Baris

Anda dapat menggunakan pernyataan `SELECT` untuk menampilkan kolom-kolom tertentu dari suatu tabel dengan menentukan nama-nama kolom, yang dipisahkan dengan koma. Contoh pada slide menampilkan semua nomor departemen dan nomor lokasi dari tabel `DEPARTMENTS`.

Pada klausa `SELECT`, ditentukan kolom-kolom yang Anda inginkan, sesuai urutan dimana Anda ingin mereka muncul pada output. Sebagai contoh, untuk menampilkan lokasi sebelum nomor departemen, dari kiri ke kanan, Anda gunakan pernyataan berikut :

```
SELECT location_id, department_id  
FROM departments;
```

LOCATION_ID	DEPARTMENT_ID
1700	10
1800	20
1500	50
. . .	

8 rows selected.

## Menulis Pernyataan-Pernyataan SQL

- Pernyataan-pernyataan SQL adalah tidak *case-sensitive*.
- Pernyataan-pernyataan SQL bisa satu atau beberapa baris.
- **Keywords** (kata kunci) tidak bisa di singkat atau dipisahkan dengan baris.
- Klausa-klausa biasanya ditempatkan pada baris-baris terpisah.
- Indent-indent (=tab) digunakan untuk menjaga agar mudah dibaca.
- Dalam *iSQL\*Plus*, pernyataan-pernyataan SQL bisa secara optional dihentikan dengan suatu titik koma (; - *semicolon*). Titik koma diperlukan jika Anda mengeksekusi beberapa pernyataan SQL.
- Dalam *SQL\*Plus*, Anda diminta untuk mengakhiri setiap pernyataan SQL dengan suatu titik koma (; - *semicolon*).

ORACLE

Copyright © 2004, Oracle. All rights reserved.

## Menulis Pernyataan-Pernyataan SQL

Gunakan aturan-aturan dan pedoman-pedoman sederhana berikut, Anda dapat membuat suatu pernyataan yang valid yang keduanya dapat mudah dibaca dan mudah diedit :

- Pernyataan SQL tidak *case-sensitive* (kecuali ditentukan).
- Pernyataan SQL dapat dituliskan pada satu atau beberapa baris.
- *Keyword* tidak dapat disingkat atau dipisahkan oleh baris.
- Klausa-klausa biasanya diletakkan pada baris yang terpisah agar mudah dibaca dan mudah diedit.
- *Indents* sebaiknya digunakan untuk membuat kode lebih mudah dibaca.
- *Keyword* biasanya ditulis dengan huruf besar; semua kata-kata lain, seperti nama-nama tabel dan kolom, ditulis dengan huruf kecil.

### Mengeksekusi Pernyataan-Pernyataan SQL

Dengan menggunakan *iSQL\*Plus*, klik tombol **Execute** untuk menjalankan perintah atau perintah-perintah dalam jendela editing.

Dengan menggunakan *SQL Plus*, akhiri pernyataan SQL dengan titik koma dan tekan **Enter** untuk menjalankan perintah.

## Default Judul-Judul Kolom

- **iSQL\*Plus :**
  - Posisi (*alignment*) default judul : Ditengah
  - Tampilan default judul : Huruf besar
- **SQL\*Plus :**
  - Posisi judul kolom karakter dan tanggal di sebelah kiri (*left-aligned*)
  - Posisi judul kolom bilangan (*number*) di sebelah kanan (*right-aligned*)
  - Tampilan default judul : Huruf besar

ORACLE

Copyright © 2004, Oracle. All rights reserved.

### Default Judul Kolom

Dalam iSQL\*Plus, judul-judul kolom ditampilkan dengan huruf besar dan posisinya ditengah.

```
SELECT last_name, hire_date, salary
FROM employees;
```

LAST_NAME	HIRE_DATE	SALARY
King	17-JUN-87	24000
Kochhar	21-SEP-89	17000
De Haan	13-JAN-93	17000
Hunold	03-JAN-90	9000
Ernst	21-MAY-91	6000
.	.	.
Higgins	07-JUN-94	12000
Gietz	07-JUN-94	8300

20 rows selected

Anda dapat mengabaikan tampilan judul-judul kolom dengan suatu alias. Kolom-kolom alias dibahas dalam pelajaran selanjutnya.



## Ekspresi-Ekspresi Aritmatika

Membuat ekspresi-ekspresi pada data bilangan dan tanggal dengan menggunakan operator-operator arimatika.

Operator	Keterangan
+	Penjumlahan
-	Pengurangan
*	Perkalian
/	Pembagian

ORACLE

Copyright © 2004, Oracle. All rights reserved.

### Ekspresi-Ekspresi Aritmatika

Anda mungkin membutuhkan untuk memodifikasi data yang ditampilkan, atau Anda mungkin ingin melakukan perhitungan atau mencari bagaimana sekenario-sekenario. Semua itu dapat dilakukan dengan menggunakan ekspresi-ekspresi aritmatika. Suatu ekspresi aritmatika dapat berisi nama-nama kolom, kontanta nilai-nilai numeric, dan operator-operator aritmatika

#### Operator-Operator Aritmatika

Daftar operator-operator aritmatika pada slide adalah yang ada dalam SQL. Anda dapat menggunakan operator-operator aritmatika dalam berbagai klausa dari suatu pernyataan SQL (kecuali pada klausa FROM)

**Catatan** : Dengan tipe data DATE dan TIMESTAMP, Anda dapat menggunakan operator penjumlahan dan pengurangan saja.

## Menggunakan Operator-Operator Aritmatika

```
SELECT last_name, salary, salary + 300
FROM employees;
```

LAST_NAME	SALARY	SALARY+300
King	24000	24300
Kochhar	17000	17300
De Haan	17000	17300
Hunold	9000	9300
Ernst	6000	6300

\*\*\*  
23 rows selected.

ORACLE

Copyright © 2004, Oracle. All rights reserved.

## Menggunakan Operator-Operator Aritmatika

Contoh pada slide menggunakan operator penjumlahan untuk menghitung suatu kenaikan penghasilan sebesar \$300 untuk semua pegawai. Pada slide juga menampilkan suatu kolom SALARY+300 pada outpunya.

Sebagai catatan hasil-hasil penghitungan kolom SALARY+300 bukan merupakan kolom baru dalam tabel EMPLOYEES, dia hanya sebagai tampilan saja. Secara *default*, nama dari kolom baru berasal dari penghitungan yang di-*generate*-nya – dalam hal ini, salary+300.

**Catatan :** Server Oracle mengabaikan ruang-ruang kosong (*blank spaces*) sebelum dan sesudah operator aritmatika

### Operator yang Dikerjakan Lebih Dulu (*Precedence*)

Jika suatu operator aritmatika berisi lebih dari satu operator, perkalian dan pembagian dievaluasi lebih dulu. Jika operator-operator dalam suatu ekspresi sama peroritasnya, maka dievaluasi dari kiri ke kanan.

Anda dapat menggunakan tanda kurung untuk menjalankan ekspresi yang diapit dengan tanda kurung agar dievaluasi lebih dulu.

### Aturan-Aturan *Precedence* :

- Perkalian dan pembagian dijalankan sebelum penjumlahan dan pengurangan.
- Operator-operator yang memiliki prioritas yang sama akan dievaluasi dari kiri ke kanan
- Tanda kurung digunakan untuk mengabaikan (*override*) dari *default* operator yang dikerjakan lebih dulu atau untuk men-jelas-kan pernyataan.

## Operator yang Dikerjakan Lebih Dulu (Operator Precedence)

```
SELECT last_name, salary, 12*salary+100  
FROM employees;
```

1

LAST_NAME	SALARY	12*SALARY+100
King	24000	288100
Kochhar	17000	204100
De Haan	17000	204100
...		

20 rows selected.

```
SELECT last_name, salary, 12*(salary+100)  
FROM employees;
```

2

LAST_NAME	SALARY	12*(SALARY+100)
King	24000	289200
Kochhar	17000	205200
De Haan	17000	205200
...		

20 rows selected.

ORACLE

Copyright © 2004, Oracle. All rights reserved.

### Operator yang Dikerjakan Lebih Dulu (lanjutan)

Contoh pertama pada slide ditampilkan nama belakang, penghasilan dan penghasilan tahunan dari para pegawai. Contoh tersebut menghitung penghasilan tahunan dengan mengalikan penghasilan bulanan dengan 12, ditambah satu kali bonus sebesar \$100.

**Catatan :** Gunakan tanda kurung untuk menguatkan urutan standar dari operator yang dikerjakan lebih dulu (*precedence*) dan untuk menambah kejelasan. Sebagai contoh, ekspresi pada slide dapat dituliskan seperti  $(12 * SALARY) + 100$  tanpa mengubah hasil.

### Menggunakan Tanda Kurung (Parentheses)

Anda dapat mengabaikan (*override*) aturan-aturan operator yang dikerjakan lebih dulu (*precedence*) dengan menggunakan tanda kurung (*parentheses*) untuk menentukan urutan yang diinginkan dimana operator-operator yang akan dieksekusi.

Contoh kedua pada slide ditampilkan nama belakang, penghasilan (bulanan) dan penghasilan tahunan dari para pegawai. Contoh tersebut menghitung penghasilan tahunan sebagai berikut : menambahkan suatu bonus bulanan sebesar \$100 ke penghasilan bulanan, dan kemudian mengalikan subtotalnya dengan 12. Karena tanda kurung, penjumlahan mendapat prioritas lebih dulu daripada perkalian.

## Mendefinisikan suatu Nilai Null

- Suatu *Null* adalah suatu nilai yang tidak ada, tidak diberikan, tidak diketahui atau tidak dipakai.
- Suatu *Null* adalah tidak sama dengan nol atau suatu ruang kosong (*blank space*).

```
SELECT last_name, job_id, salary, commission_pct  
FROM employees;
```

LAST NAME	JOB ID	SALARY	COMMISSION PCT
King	AD_PRES	24000	
Kochhar	AD_VP	17000	
***			
Deke	SA_MAN	10500	.1
Abel	SA_REP	11000	.3
Taylor	SA_REP	8800	.1
***			
Gietz	AC_ACCOUNT	6300	

20 rows selected.

ORACLE

Copyright © 2004, Oracle. All rights reserved.

### Nilai-Nilai Null

Jika suatu baris tidak ada suatu nilai data pada suatu kolom tertentu, nilai tersebut disebut *null* atau berisi suatu *null*.

Suatu *null* adalah suatu nilai yang tidak ada (*unavailable*), tidak diberikan (*unassigned*), tidak diketahui (*unknown*), atau tidak pakai (*inapplicable*). Suatu *null* tidak sama dengan nol atau suatu ruang (*space*). Nol adalah suatu bilangan dan *space* adalah suatu karakter.

Kolom-kolom dari beberapa tipe data dapat berisi *null-null*. Bagaimanapun, beberapa *constraint* (NOT NULL dan PRIMARY KEY) mencegah *null-null* untuk digunakan dalam kolom.

Pada kolom COMMISSION\_PCT dalam tabel EMPLOYEES, perhatikan bahwa hanya seorang sales manager dan sales representative yang mendapatkan sejumlah komisi. Pegawai-pegawai lain tidak mendapatkan komisi. Suatu *null* menunjukkan faktanya.

## Nilai-Nilai *Null* dalam Ekspresi-Ekspresi Aritmatika

Ekspresi-ekspresi aritmatika yang berisi suatu nilai *null* dievaluasi ke *null*.

```
SELECT last_name, 12*salary*commission_pct  
FROM employees;
```

LAST_NAME	12*SALARY*COMMISSION_PCT
King	
Kochhar	
...	
Lotney	25200
Abel	39600
Taylor	20640
...	
Gietz	

20 rows selected.

ORACLE

Copyright © 2004, Oracle. All rights reserved.

### Nilai-Nilai *Null* dalam Ekspresi-Ekspresi Aritmatika

Jika beberapa nilai kolom dalam ekspresi aritmatika adalah *null*, hasilnya adalah *null*. Sebagai contoh, jika Anda mencoba untuk melakukan pembagian dengan nol, Anda akan mendapatkan *error*. Bagaimanapun, jika Anda membagi suatu bilangan dengan *null*, hasilnya akan *null* atau tidak diketahui.

Contoh pada slide, King tidak mendapatkan suatu komisi. Karena kolom `COMMISSION_PCT` dalam ekspresi aritmatika adalah *null*, maka hasilnya adalah *null*.

Untuk informasi lebih lanjut lihat “Basic Elements of SQL” dalam *SQL Reference*.

## Mendefinisikan suatu Kolom-Kolom Alias

Suatu kolom-kolom alias :

- Memberi nama judul kolom lagi
- Berguna pada perhitungan-perhitungan
- Mengikuti setelah nama kolom (Bisa juga menggunakan opsi *keyword AS* diantara nama kolom dan alias.)
- Memerlukan tanda petik ganda jika terdapat jarak (*spaces*) atau karakter-karakter khusus atau jika *case-sensitive*

ORACLE

Copyright © 2004, Oracle. All rights reserved.

### Kolom-Kolom Alias

Ketika menampilkan hasil suatu query, secara normal *iSQL\*Plus* menggunakan nama dari kolom yang dipilih sebagai judul kolom. Judul ini mungkin tidak deskriptif dan sulit untuk dimengerti. Anda dapat merubah suatu judul kolom dengan menggunakan suatu kolom alias.

Tentukan alias setelah nama kolom dalam daftar `SELECT` menggunakan suatu spasi atau jarak sebagai suatu pemisah. Secara *default*, judul-judul alias muncul dalam bentuk huruf besar. Jika alias berisi spasi-spasi atau karakter-karakter khusus (seperti # atau \$) , atau judul aliasnya *case-sensitive*, apit alias dalam tanda petik ganda (“”).

## Menggunakan Kolom-Kolom Alias

```
SELECT last_name AS name, commission_pct comm
FROM employees;
```

NAME	COMM
King	
Kochhar	
De Haan	
...	

20 rows selected.

```
SELECT last_name "Name", salary*12 "Annual Salary"
FROM employees;
```

Name	Annual Salary
King	288000
Kochhar	204000
De Haan	204000
...	

20 rows selected.

ORACLE

Copyright © 2004, Oracle. All rights reserved.

### Kolom-Kolom Alias (lanjutan)

Pada contoh pertama ditampilkan nama-nama dan persentase komisi dari semua pegawai. Perhatikan bahwa opsi *keyword* AS digunakan sebelum nama kolom alias. Hasil dari query adalah sama apakah menggunakan keyword AS atau tidak. Perhatikan juga bahwa pernyataan SQL memiliki kolom-kolom alias, name dan comm, dalam huruf kecil, sebaliknya hasil query menampilkan judul-judul kolom dengan huruf besar. Seperti disebutkan pada slide sebelumnya, secara *default* judul-judul kolom muncul dalam huruf besar.

Pada contoh kedua ditampilkan nama belakang dan penghasilan tahunan dari semua pegawai. Karena Annual Salary berisi suatu spasi, Annual Salary harus diapit dalam tanda petik ganda. Perhatikan bahwa judul kolom pada outputnya sama persis dengan kolom alias.

## Operator Concatenation

Suatu operator *concatenation* (penggabungan) :

- Menghubungkan kolom-kolom atau rangkaian karakter dengan kolom lain
- Ditunjukkan dengan dua garis vertikal ( || )
- Membuat suatu kolom bersambung yang menunjukkan ekspresi karakter

```
SELECT last_name || job_id AS "Employees"
FROM employees;
```



The screenshot shows the output of the SQL query. The title is "Employees". The results are displayed in a table with a light green background. The first three rows are: KingAD\_PRES, KochharAD\_VP, and De HaanAD\_VP. There are three dots between the first and second columns, indicating that the output is concatenated. Below the table, it says "20 rows selected."

ORACLE

Copyright © 2004, Oracle. All rights reserved.

### Operator Concatenation (Penggabungan)

Anda dapat menghubungkan kolom-kolom dengan kolom-kolom lain, ekspresi-ekspresi aritmatika atau nilai-nilai konstanta untuk membuat suatu ekspresi karakter dengan menggunakan *operator concatenation* ( || ). Kolom-kolom di kedua sisi dari operator dikombinasikan untuk membuat suatu output kolom tunggal.

Pada contoh, LAST\_NAME dan JOB\_ID di *-concatenation* (digabungkan) dan diberi alias Employees. Perhatikan bahwa nama belakang pegawai dan kode jabatan dikombinasikan untuk membuat output kolom tunggal.

*Keyword* AS sebelum nama alias membuat klausa SELECT mudah dibaca.

#### Nilai-Nilai Null pada Operator Concatenation

Jika Anda menggabungkan suatu nilai *null* pada suatu rangkaian (*string*) karakter, hasilnya adalah suatu rangkaian karakter.

LAST\_NAME || NULL hasilnya LAST\_NAME.



## Rangkaian Karakter-Karakter Literal

- Suatu literal adalah suatu karakter, suatu bilangan, atau suatu tanggal yang disertakan pada pernyataan `SELECT`.
- Nilai-nilai literal tanggal dan karakter harus diapit dengan tanda petik tunggal.
- Setiap rangkaian karakter dikeluarkan sekali untuk setiap baris yang dikembalikan.

ORACLE

Copyright © 2004, Oracle. All rights reserved.

## Rangkaian Karakter-Karakter Literal

Suatu literal adalah suatu karakter, suatu bilangan, atau suatu tanggal yang ada pada daftar `SELECT` dan itu bukan suatu nama kolom atau suatu kolom alias. Literal dicetak untuk setiap baris yang dikembalikan. Rangkaian literal suatu teks dengan format bebas dapat disertakan pada hasil query dan diperlakukan sama seperti suatu kolom dalam daftar `SELECT`.

Literal-literal tanggal dan karakter *harus* diapit dengan tanda petik tunggal ( ` ` ); literal-literal bilangan tidak perlu diapit dengan tanda petik tunggal.

## Menggunakan Rangkaian Karakter-Karakter Literal

```
SELECT last_name || ' is a ' || job_id  
       AS "Employee Details"  
FROM   employees;
```

Employee Details	
King	is a AD_PRES
Kochhar	is a AD_VP
De Haan	is a AD_VP
Hunold	is a IT_PROG
Ernst	is a IT_PROG
Lorentz	is a IT_PROG
Mourgos	is a ST_MAN
Rajs	is a ST_CLERK
...	
20 rows selected.	

ORACLE

Copyright © 2004, Oracle. All rights reserved.

### Rangkaian Karakter-Karakter Literal (lanjutan)

Contoh pada slide menampilkan nama belakang dan kode jabatan semua pegawai. Kolomnya diberi judul **Employee Details**. Perhatikan bahwa spasi-spasi diantara tanda perik tunggal pada pernyataan `SELECT`. Spasi-spasi menambah output mudah untuk dibaca.

Pada contoh berikut, nama belakang (*last name*) dan penghasilan (*salary*) untuk setiap pegawai di-*concatenate* dengan suatu literal untuk mengembalikan baris-baris yang lebih berarti :

```
SELECT last_name || ' : 1 Motnh salary = ' || salary Monthly  
FROM employees;
```

MONTHLY	
King	: 1 Month salary = 24000
Kochhar	: 1 Month salary = 17000
De Haan	: 1 Month salary = 17000
Hunold	: 1 Month salary = 9000
Ernst	: 1 Month salary = 6000
Lorentz	: 1 Month salary = 4200
Mourgos	: 1 Month salary = 5800
Rajs	: 1 Month salary = 3500

...  
20 rows selected.

## Operator *Alternative Quote* (q)

- Menentukan pemisah tanda petik yang Anda buat
- Memilih beberapa pemisah
- Menambah kemudahan baca dan penggunaan

```
SELECT department name ||  
       q'[, it's assigned Manager Id: ]'  
       || manager_id  
       AS "Department and Manager"  
FROM departments;
```

```
Department and Manager  
Administration, it's assigned manager ID: 200  
Marketing, it's assigned manager ID: 201  
Shipping, it's assigned manager ID: 124  
***  
8 rows selected.
```

ORACLE

Copyright © 2004, Oracle. All rights reserved.

## Operator *Alternative Quote* (q)

Beberapa pernyataan SQL menggunakan karakter-karakter literal dalam ekspresi-ekspresi atau kondisi-kondisi. Jika literal itu sendiri berisi tanda petik tunggal, Anda dapat menggunakan operator *quote* (q) dan memilih pemisah tanda petik yang Anda buat.

Anda dapat memilih beberapa pemisah yang sesuai, *single-byte* atau *multibyte* atau beberapa pasang karakter berikut : [ ], { }, ( ) atau <> .

Pada contoh ditunjukkan, rangkaian berisi suatu tanda petik tunggal, yang secara normal ditafsirkan sebagai pemisah dari suatu rangkaian karakter. Dengan menggunakan operator q, bagaimanapun tanda kurung [ ] digunakan sebagai pemisah tanda petik. Rangkaian diantara pemisah tanda kurung ditafsirkan sebagai suatu rangkaian karakter literal.

## Baris-Baris yang Sama

Tampilan *default* dari query-query adalah baris-baris, termasuk baris-baris yang sama.

```
SELECT department_id
FROM employees;
```

1

DEPARTMENT_ID
90
90
90
...

20 rows selected.

```
SELECT DISTINCT department_id
FROM employees;
```

2

DEPARTMENT_ID
10
20
60
...

8 rows selected.

ORACLE

Copyright © 2004, Oracle. All rights reserved.

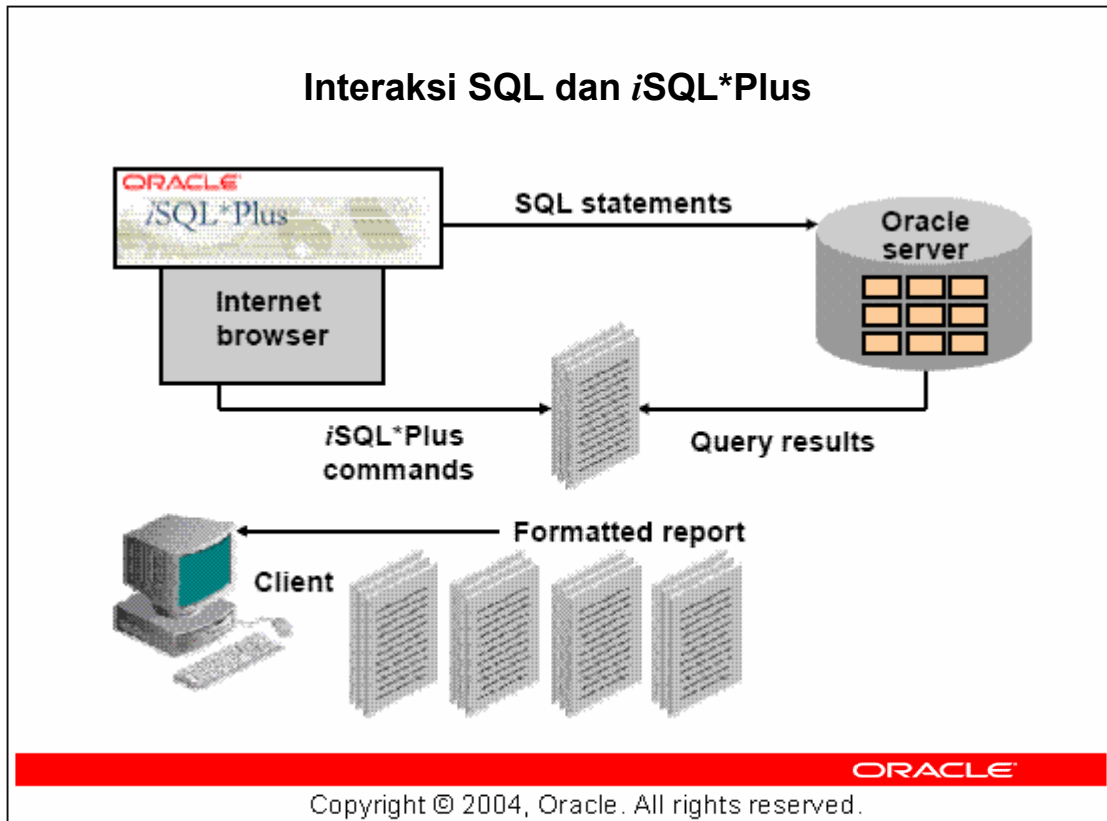
## Baris-Baris yang Sama

iSQL\*Plus akan menampilkan hasil-hasil suatu query tanpa membuang baris-baris yang sama, kecuali jika Anda menyatakan lain. Pada slide contoh pertama menampilkan semua nomor departemen dari tabel EMPLOYEES. Perhatikan bahwa nomor-nomor departemen dihasilkan berulang kali.

Untuk membuang baris-baris yang sama dalam hasil query, sertakan *keyword* DISTINCT pada klausa SELECT segera setelah *keyword* SELECT. Pada slide contoh kedua, tabel EMPLOYEES sebenarnya berisi 20 baris, tapi hanya 7 nomor departemen yang unik dalam tabel yang ditampilkan.

Anda dapat menentukan beberapa kolom setelah *qualifier* DISTINCT. *Qualifier* DISTINCT berakibat pada seluruh kolom yang dipilih, dan menghasilkan setiap kombinasi yang berbeda (*distinct*) dari kolom-kolom.

```
SELECT DISTINCT department_id, job_id
FROM employees;
```



## SQL dan iSQL\*Plus

SQL adalah *command language* untuk komunikasi pada server Oracle dari berbagai tool atau aplikasi. Oracle SQL berisi banyak perluasan (*extension*).

iSQL\*Plus adalah suatu tool Oracle yang mengenali dan mengirimkan pernyataan-pernyataan SQL ke server Oracle untuk eksekusi dan berisi *command language* Oracle sendiri.

### Fitur-Fitur SQL

- Dapat digunakan oleh beberapa user, termasuk mereka yang sedikit atau tidak berpengalaman dalam pemrograman
- Adalah suatu bahasa *nonprocedural*
- Seperti bahasa Inggris

### Fitur-Fitur iSQL\*Plus

- Diakses dari suatu browser
- Menerima pernyataan SQL
- Menyediakan editing online untuk memodifikasi pernyataan-pernyataan SQL
- Pengaturan control-kontrol *environmental*
- Hasil-hasil format query ke dalam suatu laporan dasar
- Akses lokal dan *remote databases*

## Pernyataan-Pernyataan SQL Versus Perintah-Perintah *iSQL\*Plus*

### SQL

- Suatu bahasa
- Standar ANSI
- Kata kunci tidak bisa disingkat
- Pernyataan-pernyataan perubahan data dan definisi-definisi tabel dalam database

SQL  
statements

### *iSQL\*Plus*

- Suatu lingkungan (*environment*)
- *Oracle-proprietary* (milik Oracle)
- Kata kunci bisa disingkat
- Perintah-perintah tidak mengijinkan perubahan dari nilai-nilai dalam database
- Berjalan pada suatu *browser*
- Dijalankan secara terpusat; tidak harus diterapkan pada setiap mesin

*iSQL\*Plus*  
commands

**ORACLE**

Copyright © 2004, Oracle. All rights reserved.

### SQL dan *iSQL\*Plus* (lanjutan)

Berikut adalah tabel perbandingan antara SQL dan *iSQL\*Plus* :

SQL	<i>iSQL*Plus</i>
Adalah suatu bahasa untuk komunikasi dengan server Oracle untuk akses data	Mengenali pernyataan-pernyataan SQL dan mengirimnya ke server
Adalah berdasarkan standar SQL American National Standards Institute (ANSI)	Adalah penghubung milik Oracle ( <i>Oracle-proprietary</i> ) untuk mengeksekusi pernyataan-pernyataan SQL
Mengambil Data, memanipulasi data dan pendefinisian tabel dalam database	Tidak mengijinkan manipulasi nilai-nilai dalam database
Tidak memiliki karakter penghubung	Memiliki suatu <i>dash</i> (-) sebagai karakter penghubung jika perintah lebih dari satu baris
Tidak dapat disingkat	Dapat disingkat
Menggunakan fungsi-fungsi untuk melakukan suatu format	Menggunakan perintah-perintah untuk format data

## Gambaran dari *iSQL\*Plus*

Setelah Anda *log in* ke *iSQL\*Plus*, Anda bisa :

- Melihat struktur-struktur tabel
- Memasukkan, mengeksekusi dan mengedit pernyataan-pernyataan SQL
- Menyimpan atau menambah pernyataan-pernyataan SQL ke file
- Mengeksekusi atau mengedit pernyataan-pernyataan yang disimpan dalam file-file *script* yang disimpan

ORACLE

Copyright © 2004, Oracle. All rights reserved.

### *iSQL\*Plus*

*iSQL\*Plus* adalah suatu *environment* (lingkungan) yang didalamnya dapat dilakukan sebagai berikut :

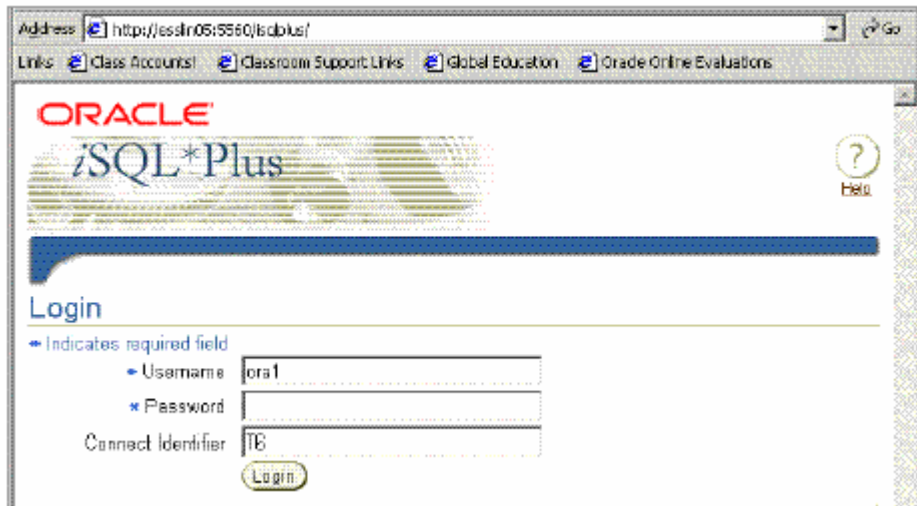
- Mengeksekusi pernyataan-pernyataan SQL untuk *me-retrieve* (menggambil), memodifikasi, menambah, dan menghapus data dari database
- Memformat, melakukan perhitungan, menyimpan, dan mencetak hasil-hasil query dalam suatu laporan
- Membuat *script* untuk menyimpan pernyataan-pernyataan SQL untuk digunakan kembali

Perintah-perintah *iSQL\*Plus* dapat dibagi kedalam kategori utama sebagai berikut :

Kategori	Tujuan
<i>Environment</i>	Secara umum berakibat pada perilaku pernyataan-pernyataan SQL pada suatu sesi
<i>Format</i>	Format hasil-hasil query
<i>File Manipulation</i>	Menyimpan pernyataan-pernyataan dalam suatu file-file <i>script</i> dan menjalankan pernyataan-pernyataan dari file-file <i>script</i>
<i>Execution</i>	Mengirim pernyataan-pernyataan SQL dari <i>browser</i> ke server Oracle
<i>Edit</i>	Memodifikasi pernyataan-pernyataan SQL ke jendela Edit
<i>Interaction</i>	Memungkinkan Anda untuk membuat dan menjalankan variable-variable ke pernyataan-pernyataan SQL, mencetak nilai-nilai variable, dan mencetak pesan-pesan ( <i>messages</i> ) layar
<i>Miscellaneous</i>	Memiliki beragam perintah untuk menghubungkan ke database, memanipulasi <i>iSQL*Plus environment</i> dan menampilkan definisi-definisi kolom

## Logging In ke iSQL\*Plus

Dari lingkungan *browser* Anda :



The screenshot shows a web browser window with the address bar containing 'http://esslin05:5560/lsqplus/'. The page features the Oracle logo and 'iSQL\*Plus' text. Below the header, there is a 'Login' section with a legend indicating required fields. The 'Username' field contains 'ora1', the 'Password' field is empty, and the 'Connect Identifier' field contains 'T8'. A 'Login' button is located below the fields. A 'Help' link is visible in the top right corner.

ORACLE

Copyright © 2004, Oracle. All rights reserved.

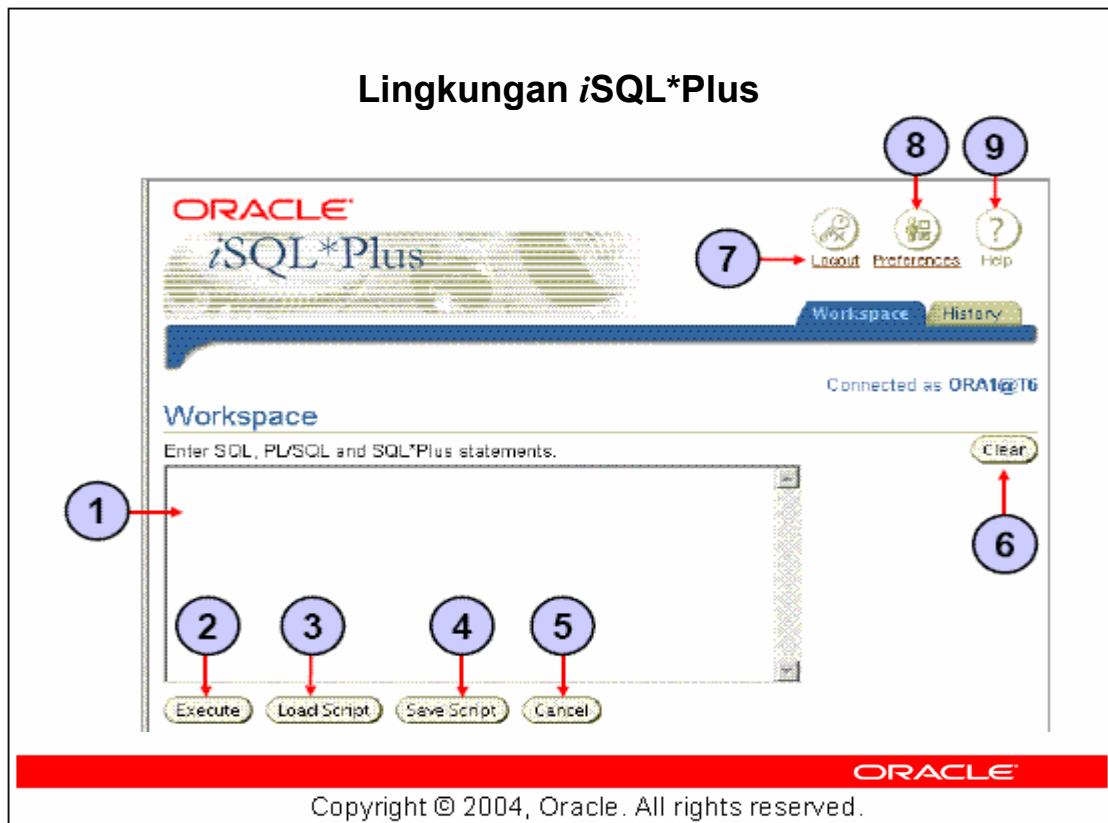
### Logging In ke iSQL\*Plus

Untuk *Log in* dari suatu lingkungan *browser*:

1. Mulai *browser*.
2. Masukkan alamat URL dari lingkungan *iSQL\*Plus*.
3. Pada halaman *Login*, masukkan nilai-nilai yang sesuai pada *Username*, *Password* dan kotak *Connect Identifier*.



## Lingkungan iSQL\*Plus



### Lingkungan (*Environment*) iSQL\*Plus

Pada *browser*, halaman *workspace* iSQL\*Plus terdapat beberapa area kunci :

1. **Text Box** : Area dimana Anda menuliskan pernyataan-pernyataan SQL dan perintah-perintah iSQL\*Plus
2. **Execute Button** : Klik untuk mengeksekusi pernyataan-pernyataan dan perintah-perintah yang ada dalam *text box*
3. **Load Script Button** : Mengangkat suatu *form* dimana Anda dapat mengidentifikasi suatu *path* dan nama file atau suatu URL yang berisi SQL, PL/SQL, atau perintah-perintah SQL\*Plus dan me-load mereka ke dalam *text box*
4. **Save Script Button** : Menyimpan isi dari *text box* ke suatu file
5. **Cancel Button** : Menghentikan eksekusi perintah dalam *text box*
6. **Clear Screen Button** : Klik untuk membersihkan teks dalam *text box*
7. **Logout Icon** : Klik untuk mengakhiri sesi iSQL\*Plus dan kembali ke halaman login iSQL\*Plus
8. **Preferences Icon** : Klik untuk mengganti konfigurasi *interface* Anda, konfigurasi sistem atau *password*
9. **Help Icon** : Menyediakan akses ke dokumentasi *help* iSQL\*Plus

## Menampilkan Struktur Tabel

Gunakan perintah `DESCRIBE` *iSQL\*Plus* untuk menampilkan struktur dari suatu tabel :

```
DESC[RIBE] tablename
```

ORACLE

Copyright © 2004, Oracle. All rights reserved.

### Menampilkan Struktur Tabel

Dalam *iSQL\*Plus*, Anda dapat menampilkan struktur tabel dengan menggunakan perintah `DESCRIBE`. Perintah menampilkan nama-nama kolom dan tipe data-tipe data, serta menampilkan apakah suatu kolom *harus* berisi data (yaitu, apakah kolom memiliki suatu *constraint* `NOT NULL`).

Dalam sintak, *tablename* adalah nama tabel dari beberapa tabel yang ada, *view* atau *synonym* yang dapat diakses oleh user.

## Menampilkan Struktur Tabel

```
DESCRIBE employees
```

Name	Null?	Type
EMPLOYEE_ID	NOT NULL	NUMBER(6)
FIRST_NAME		VARCHAR2(20)
LAST_NAME	NOT NULL	VARCHAR2(25)
EMAIL	NOT NULL	VARCHAR2(25)
PHONE_NUMBER		VARCHAR2(20)
HIRE_DATE	NOT NULL	DATE
JOB_ID	NOT NULL	VARCHAR2(10)
SALARY		NUMBER(8,2)
COMMISSION_PCT		NUMBER(2,2)
MANAGER_ID		NUMBER(6)
DEPARTMENT_ID		NUMBER(4)

ORACLE

Copyright © 2004, Oracle. All rights reserved.

### Menampilkan Struktur Tabel (lanjutan)

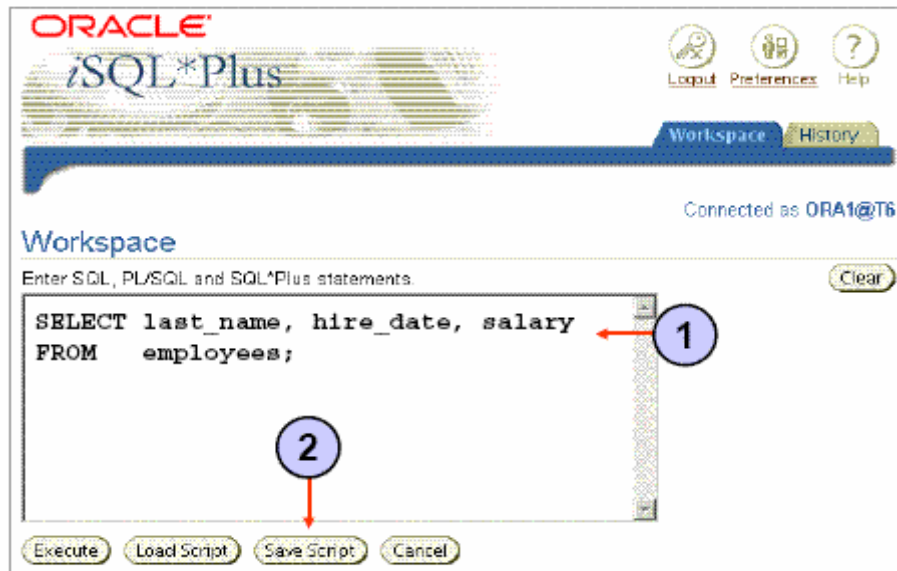
Contoh pada slide menampilkan informasi tentang struktur dari tabel EMPLOYEES.

Pada hasilnya ditampilkan, *null?* menunjukkan bahwa nilai-nilai untuk kolom tersebut mungkin tidak diketahui. NOT NULL menunjukkan bahwa suatu kolom harus berisi data. *Type* menampilkan tipe data untuk suatu kolom.

Tipe data-tipe data dijelaskan pada tabel berikut ini :

Tipe Data	Keterangan
NUMBER ( <i>p</i> , <i>s</i> )	Nilai angka yang memiliki suatu jumlah digit maksimum <i>p</i> , dengan digit <i>s</i> disebelah kanan untuk desimal
VARCHAR2 ( <i>s</i> )	Nilai karakter <i>variable-length</i> dengan ukuran maksimum <i>s</i>
DATE	Nilai tanggal dan waktu antara January 1, 4712 B.C dan December 31,9999 A.D.
CHAR ( <i>s</i> )	Nilai karakter <i>fixed-length</i> dengan ukuran <i>s</i>

## Berinteraksi dengan File-File Script



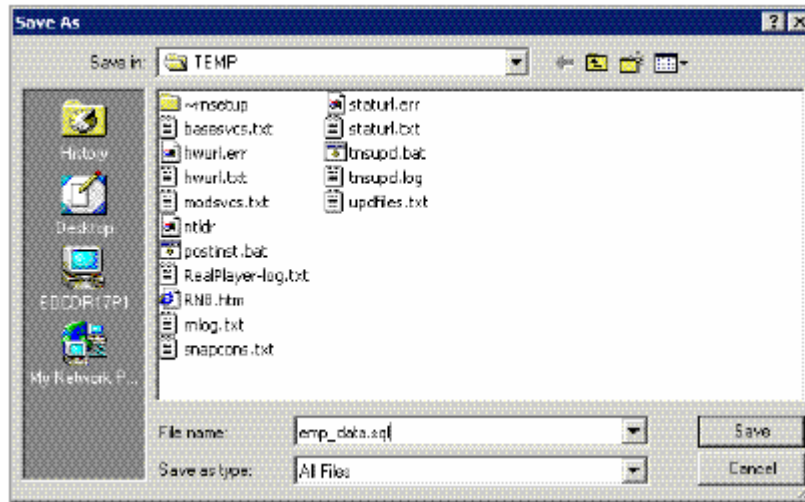
### Berinteraksi dengan File-File Script

#### Menempatkan Pernyataan-pernyataan dan Perintah-Perintah kedalam suatu File Script

Anda dapat menyimpan perintah-perintah dan pernyataan-pernyataan dari *text box* dalam *iSQL\*Plus* ke suatu file *text script* file sebagai berikut :

1. Tulis pernyataan-pernyataan SQL dalam text box *iSQL\*Plus*.
2. Klik tombol **Save Script**. Hal ini membuka *dialog box Windows File Save*. Beri nama file. Default ekstensi file adalah `.uix`. Anda dapat mengganti tipe file ke file text atau disimpan sebagai suatu file `.sql`.

## Berinteraksi dengan File-File *Script*



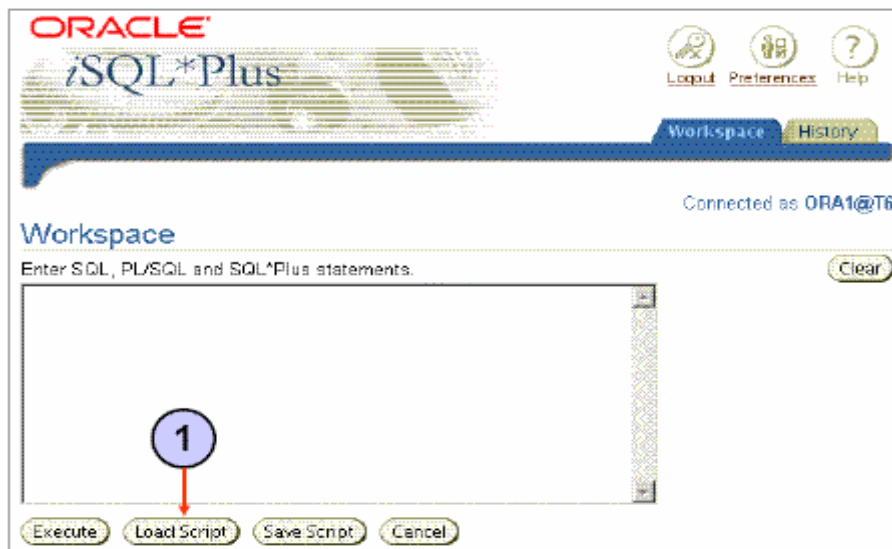
ORACLE

Copyright © 2004, Oracle. All rights reserved.

### Berinteraksi dengan File-File *Script* (lanjutan)

Pada contoh ditampilkan, pernyataan `SELECT` ditulis dalam *text box* yang disimpan ke suatu file dengan nama `emp_data.sql`. Anda dapat memilih tipe file, nama file, dan lokasi dimana Anda akan menyimpan file *script*.

## Berinteraksi dengan File-File *Script*



ORACLE

Copyright © 2004, Oracle. All rights reserved.

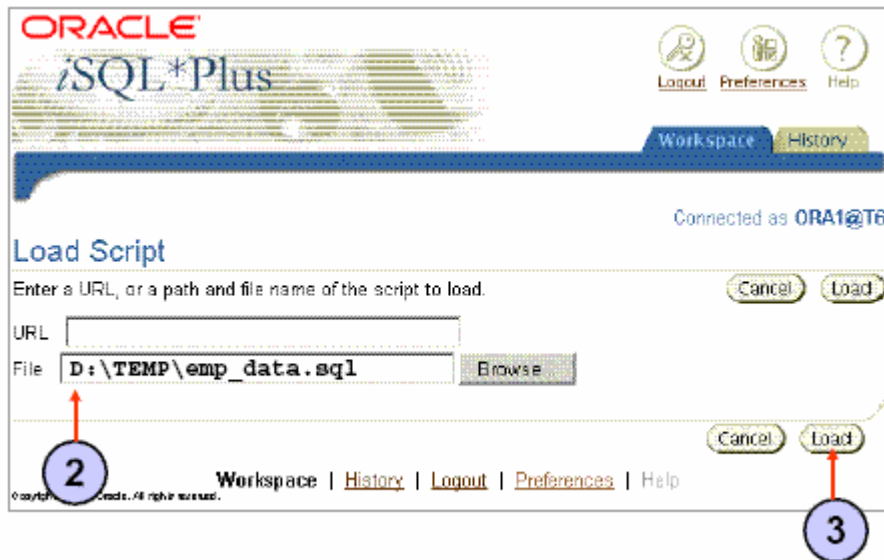
### Berinteraksi dengan File-File *Script* (lanjutan)

#### Menggunakan Pernyataan-Pernyataan dan Perintah-Perintah dari suatu File *Script* dalam *iSQL\*Plus*

Anda dapat menggunakan perintah-perintah dan pernyataan-pernyataan yang disimpan sebelumnya dari suatu file *script* dalam *iSQL\*Plus* sebagai berikut :

1. Klik tombol **Load Script**. Ini akan membuka suatu *form* dimana Anda dapat menulis nama file atau suatu URL yang berisi perintah-perintah SQL, PL/SQL, atau SQL\*Plus yang Anda ingin masukkan dalam *text box*.

## Berinteraksi dengan File-File *Script*



### Berinteraksi dengan File-File *Script* (lanjutan)

- 2 Masukkan nama *script* dan *path*, atau lokasi URL. Atau Anda dapat meng-klik tombol **Browse** untuk mencari nama *script* dan lokasinya
- 3 Klik tombol **Load** untuk mengambil isi file atau lokasi URL ke dalam *text box*.

## Halaman History iSQL\*Plus

The screenshot displays the iSQL\*Plus interface. At the top, there are tabs for 'Workspace' and 'History', with 'History' being the active tab. Below the tabs, it says 'Connected as ORA1@T6'. The main area is titled 'History' and contains a message: 'The scripts listed are for the current session. Script history is not available for previous sessions.' Below this message is a 'Select scripts and ...' section with 'Delete' and 'Load' buttons. A list of SQL scripts follows, each with a checkbox. The 'Load' button and the first checkbox are highlighted with red circles and numbers 2 and 1 respectively. The 'History' tab is highlighted with a red circle and number 3. At the bottom, a red bar contains the Oracle logo and the text 'ORACLE Copyright © 2004, Oracle. All rights reserved.'

### Menjalankan Pernyataan-Pernyataan Sebelumnya

Halaman *History* dalam iSQL\*Plus memungkinkan Anda untuk mengeksekusi pernyataan-pernyataan sebelumnya pada sesi Anda. Halaman *History* menampilkan pernyataan-pernyataan SQL dan perintah-perintah iSQL\*Plus yang sering Anda gunakan.

Untuk mengeksekusi kembali pernyataan-pernyataan :

1. Pilih pernyataan yang akan Anda eksekusi.
2. Klik tombol *Load*.

#### Catatan

- Anda dapat mengontrol jumlah dari pernyataan-pernyataan yang akan ditampilkan pada halaman *History* dengan mengatur *Preferences*.
- Anda dapat memilih untuk menghapus pernyataan-pernyataan terpilih dengan mengklik tombol *Delete*.



## Halaman History iSQL\*Plus



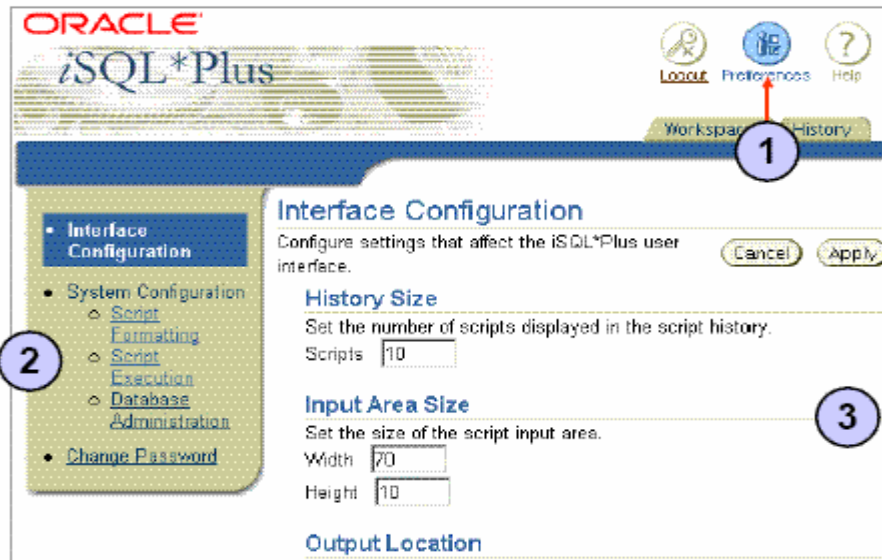
ORACLE

Copyright © 2004, Oracle. All rights reserved.

### Menjalankan Pernyataan-Pernyataan Sebelumnya (lanjutan)

2. Kembali ke halaman *Workspace*.
3. Klik tombol *Execute* untuk menjalankan perintah-perintah yang telah di-load ke dalam *text box*.

## Mengatur *iSQL\*Plus Preferences*



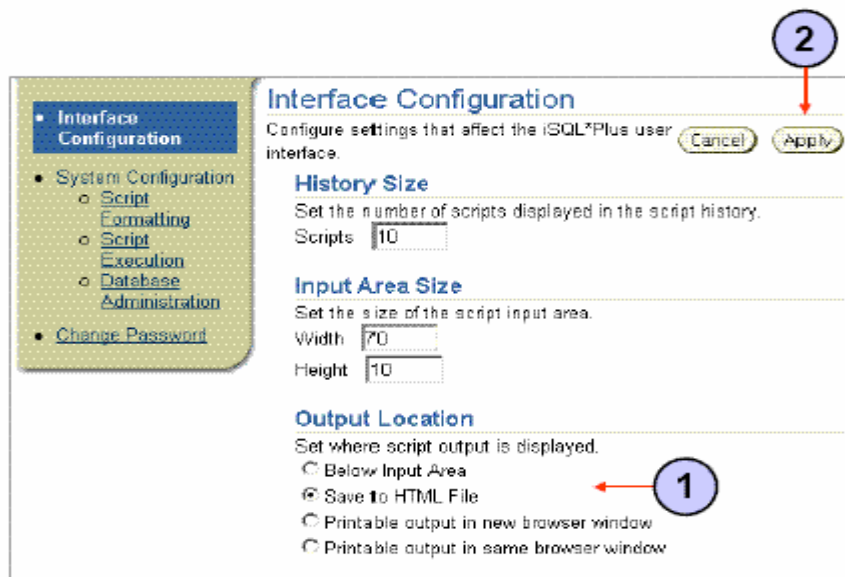
ORACLE

Copyright © 2004, Oracle. All rights reserved.

### ***iSQL\*Plus Preference***

- Anda dapat mengatur *preferences* untuk *iSQL\*Plus* Anda dengan mengklik icon *Preferences*.
- *Preferences* dikelompokkan kedalam kategori-kategori. Anda dapat mengatur *preferences* untuk *script formatting*, *script executing*, dan *database administration*, serta Anda dapat mengganti *password*.
- Ketika Anda memilih suatu kategori *preference*, suatu *form* muncul yang menyuruh Anda untuk mengatur *preference* untuk kategori tersebut.

## Mengatur *Preference* Lokasi Output



ORACLE

Copyright © 2004, Oracle. All rights reserved.

### Merubah Lokasi Output

Anda dapat mengirimkan hasil-hasil yang telah di-generate oleh suatu pernyataan SQL atau perintah *iSQL\*Plus* ke layar (*default*), suatu file atau ke jendela *browser* lainnya.

Pada halaman *Preference* :

1. Pilih **Output Location**.
2. Klik tombol **Apply**.

## Ringkasan

Dalam pelajaran ini, Anda sudah belajar bagaimana untuk :

- Menulis suatu pernyataan **SELECT** yang :
  - Mengembalikan semua baris dan kolom dari suatu tabel
  - Mengembalikan kolom-kolom tertentu dari suatu tabel
  - Menggunakan kolom-kolom alias untuk menampilkan keterangan judul-judul kolom
- Menggunakan lingkungan **iSQL\*Plus** untuk menulis, menyimpan dan mengeksekusi pernyataan-pernyataan SQL dan perintah-perintah **iSQL\*Plus**

```
SELECT *|{[DISTINCT] column/expression [alias],...}  
FROM table;
```

ORACLE

Copyright © 2004, Oracle. All rights reserved.

### Pernyataan SELECT

Dalam pelajaran ini, Anda sudah mempelajari bagaimana cara untuk mengambil (*retrieve*) data dari tabel database dengan pernyataan **SELECT**.

```
SELECT * | {[DISTINCT] column [alias], . . . }  
FROM table;
```

Dalam sintak :

SELECT	adalah suatu daftar dari satu atau beberapa kolom
*	menampilkan semua kolom
DISTINCT	menghilangkan duplikasi
column / expression	memilih nama kolom atau ekspresi
Alias	memberikan nama kolom yang dipilih dengan judul yang berbeda
FROM table	Menentukan isi-isi kolom dari suatu tabel

### iSQL\*Plus

**iSQL\*Plus** adalah suatu *environment* (lingkungan) pengeksekusian yang dapat Anda gunakan untuk mengirimkan pernyataan-pernyataan SQL ke server database dan untuk mengedit dan menyimpan pernyataan-pernyataan SQL. Pernyataan-pernyataan dapat dieksekusi dari *prompt* SQL atau dari suatu file *script*.