

PRACTICE 1

1. Initiate an *iSQL*Plus* session using the user ID and password that are provided by the instructor (pada pengerjaan tugas ini kami login dengan user ID=HR dan password=1234)
2. *iSQL*Plus* commands access the database (**False**)
3. Menampilkan nama belakang, id pekerjaan dan gaji yang diberi nama (alias) "SAL" dari tabel employees.

```
SELECT last_name, job_id, salary AS Sal FROM employees; (True)
```

LAST_NAME	JOB_ID	SAL
Gates	SH_CLERK	2900
Perkins	SH_CLERK	2500
Bell	SH_CLERK	4000

107 rows selected

4. `SELECT * FROM job_grades;` (**True**)
Menampilkan semua value yang ada di tabel job_grades
5. `SELECT employee_id, last_name sal x 12 ANNUAL SALARY FROM employees;`
 - ✓ Tabel employees tidak memiliki kolom bernama "sal" yang ada "salary".
 - ✓ Operator yang dipakai salah bukan X tetapi *
 - ✓ Penulisan alias salah, tidak boleh mengandung spasi terkecuali diapit oleh kutip, seharusnya ANNUAL_SALARY atau "ANNUAL SALARY"
 - ✓ Tanda koma tidak ada setelah penulisan last_name.Jadi secara keseluruhan statement ini salah.

6. Mendeskripsikan struktur table departments

```
DESC departments;
```

Name	Null?	Type
DEPARTMENT_ID	NOT NULL	NUMBER(4)
DEPARTMENT_NAME	NOT NULL	VARCHAR2(30)
MANAGER_ID		NUMBER(6)
LOCATION_ID		NUMBER(4)

Menampilkan semua value/konten dari table departments

```
SELECT * FROM departments;
```

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
10	Administration	200	1700
20	Marketing	201	1800
30	Purchasing	114	1700

7. Mendeskripsikan struktur dari table employees

DESC 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)

Menampilkan id pekerja, nama belakang, id pekerjaan, tanggal mulai bekerja dari table employees.

```
SELECT employee_id, last_name, job_id, hire_date StartDate
FROM employees;
```

8. Menampilkan id pekerja, nama belakang, id pekerjaan, tanggal mulai bekerja dari table employees.

```
SELECT employee_id, last_name, job_id, hire_date StartDate
FROM employees;
```

EMPLOYEE_ID	LAST_NAME	JOB_ID	STARTDATE
198	OConnell	SH_CLERK	21-JUN-99
199	Grant	SH_CLERK	13-JAN-00
200	Whalen	AD_ASST	17-SEP-87

107ws selected

9. Menampilkan job_id dari table employees tanpa adanya pengulangan untuk job_id yang sama, menggunakan command DISTINCT.

```
SELECT DISTINCT job_id FROM employees;
```

```
JOB_ID
-----
AC_ACCOUNT
AC_MGR
AD_ASST
AD PRES
AD_UP
FI_ACCOUNT
FI_MGR
HR_REP
IT_PROG
MK_MAN
MK_REP
```

19 rows selected

10. Menampilkan employee_id dengan alias Emp #, last_name sebagai Employee dan seterusnya.

```
SELECT employee_id "Emp #", last_name "Employee", job_id "Job",
hire_date "Hire Date" FROM employees;
```

Emp #	Employee	Job	Hire Date
198	OConnell	SH_CLERK	21-JUN-99
199	Grant	SH_CLERK	13-JAN-00
200	Whalen	AD_ASST	17-SEP-87

107 rows selected

- Menampilkan last_name dan job_id dengan cara digabungkan, menggunakan concatenation. Dalam hal ini ||, dimana tampilannya dipisahkan oleh koma dan spasi dan diberi alis "Employee and Title".

```
SELECT last_name||', '||job_id "Employee and Title" FROM
employees;
```

```
Employee and Title
-----
OConnell, SH_CLERK
Grant, SH_CLERK
Whalen, AD_ASST
107 rows selected
```

- Menampilkan konten-konten dimana digabungkan menggunakan perintah concat (||), dan THE_OUTPUT dipakai sebagai aliasnya.

```
SELECT employee_id || ',' || first_name || ',' || last_name || ','
|| email || ',' || phone_number || ',' || job_id || ',' ||
manager_id || ',' || hire_date || ',' || salary || ',' ||
commission_pct || ',' || department_id THE_OUTPUT FROM employees;
```

```
THE_OUTPUT
-----
198,Donald,OConnell,DOCONNEL,650.507.9833,SH_CLERK,124,21-JUN-99,2600,,50
199,Douglas,Grant,DGRANT,650.507.9844,SH_CLERK,124,13-JAN-00,2600,,50
200,Jennifer,Whalen,JWHALEN,515.123.4444,AD_ASST,101,17-SEP-87,4400,,10
107 rows selected
```

PRACTICE 2

- Penggunaan klausa where, menampilkan last_name dan salary dari employees dimana gajinya lebih besar (>) dari 12000.

```
SELECT last_name, salary FROM employees WHERE salary > 12000;
```

```
LAST_NAME          SALARY
-----
Hartstein          13000
King                24000
Kochhar            17000
De Haan            17000
Russell            14000
Partners           13500
```

6 rows selected.

- Menampilkan last_name dan department_id dari employee_id=176

```
SELECT last_name, department_id
FROM employees WHERE employee_id = 176;
```

```
LAST_NAME          DEPARTMENT_ID
-----
Taylor              80
```

- Menampilkan employees yang memiliki gaji terbesar dan terkecil tetapi gaji bukan diantara \$5000-\$12000, yang berarti bisa diatas \$12000 dan dibawah \$5000.

```
SELECT last_name, salary FROM employees
WHERE salary NOT BETWEEN 5000 AND 12000;
```

LAST_NAME	SALARY
Cabrio	3000
Chung	3800
Dilly	3600
Gates	2900
Perkins	2500
Bell	4000
Everett	3900
McCain	3200
Jones	2800
Walsh	3100
Feeney	3000

55 rows selected.

- Menampilkan last_name, job_id, hire_date dari tabel employees dimana last_name adalah Matos dan Taylor, yang diurutkan berdasarkan hire_date.

```
SELECT last_name, job_id, hire_date FROM employees
WHERE last_name IN ('Matos', 'Taylor') ORDER BY hire_date;
```

LAST_NAME	JOB_ID	HIRE_DATE
Taylor	SH_CLERK	24-JAN-98
Matos	ST_CLERK	15-MAR-98
Taylor	SA_REP	24-MAR-98

- Menampilkan last_name, department_id dari tabel employees tetapi yang department_id nya=20 dan 50 saja. Diurutkan berdasarkan last_name secara ascending.

```
SELECT last_name, department_id FROM employees
WHERE department_id IN (20, 50) ORDER BY last_name ASC;
```

LAST_NAME	DEPARTMENT_ID
Uollman	50
Walsh	50
Weiss	50

47 rows selected.

- Menampilkan last_name yang diberi alias "Employee" dan salary dengan alias "Monthly Salary" dari tabel employees dimana salary diantara 5000 dan 12000 dan department_id (20, 50).

```
SELECT last_name "Employee", salary "Monthly Salary"
FROM employees
WHERE salary BETWEEN 5000 AND 12000 AND department_id IN (20, 50);
```

Employee	Monthly Salary
Fay	6000
Weiss	8000
Fripp	8200
Kaufling	7900
Uollman	6500
Hourgos	5800

6 rows selected.

7. Menampilkan last_name, hire_date dari employees dimana hire_date tahun 1994.

```
SELECT last_name, hire_date
FROM employees WHERE hire_date LIKE '%94';
```

LAST_NAME	HIRE_DATE
Mauris	07-JUN-94
Baer	07-JUN-94
Higgins	07-JUN-94
Gietz	07-JUN-94
Greenberg	17-AUG-94
Faviet	16-AUG-94
Raphaely	07-DEC-94

7 rows selected.

8. Menampilkan employees yang tidak memiliki manager (manager_id sama dengan null)

```
SELECT last_name, job_id FROM employees WHERE manager_id IS NULL;
```

LAST_NAME	JOB_ID
King	AD_PRES

9. Menampilkan last_name, salary, commission_pct dari tabel employees yang mendapatkan komisi. Urutkan berdasarkan salary dan commission_pct secara descending.

```
SELECT last_name, salary, commission_pct FROM employees
WHERE commission_pct IS NOT NULL
ORDER BY salary DESC, commission_pct DESC;
```

LAST_NAME	SALARY	COMMISSION_PCT
Johnson	6200	.1
Kumar	6100	.1

35 rows selected.

10. Menampilkan last_name dan salary untuk employees dimana besar gajinya (salary) di input terlebih dahulu melalui sebuah prompt.

```
SELECT last_name, salary FROM employees WHERE salary > &komisi;
```

```
Enter value for komisi: 12000
old 3: where salary > &komisi
new 3: where salary > 12000
```

LAST_NAME	SALARY
Hartstein	13000
King	24000
Kochhar	17000
De Haan	17000
Russell	14000
Partners	13500

6 rows selected.

11. Menampilkan last_name, employee_id, salary dan department_id dimana manager_id di input melalui prompt dan diurutkan berdasarkan oleh kolom yang kita tentukan.

```
SELECT employee_id, last_name, salary, department_id
FROM employees WHERE manager_id = &nomor ORDER BY &kolom;
```

```
Enter value for nomor: 103
old 3: WHERE manager_id = &nomor
new 3: WHERE manager_id = 103
Enter value for kolom: salary
old 4: ORDER BY &kolom
new 4: ORDER BY salary
```

EMPLOYEE_ID	LAST_NAME	SALARY	DEPARTMENT_ID
107	Lorentz	4200	60
105	Austin	4800	60
106	Pataballa	4800	60
104	Ernst	3900	60

12. Menampilkan last_name dimana huruf ke-3 = a

```
SELECT last_name FROM employees WHERE last_name LIKE '___a%';
```

```
LAST_NAME
-----
Grant
Grant
Whalen
```

13. Menampilkan last_name yang mengandung huruf a dan e

```
SELECT last_name FROM employees
WHERE last_name LIKE '%a%' AND last_name LIKE '%e%';
```

```
LAST_NAME
-----
Partners
Patel
Philtanker
Raphaely
Sewall
Whalen

17 rows selected.
```

14. Menampilkan last_name, job_id, salary dimana hanya yang memiliki job_id = SA_REP, ST_CLERK dan salary bukan = 2500, 3500, 7000.

```
SELECT last_name, job_id, salary FROM employees
WHERE job_id IN ('SA_REP', 'ST_CLERK')
AND salary NOT IN (2500, 3500, 7000);
```

LAST_NAME	JOB_ID	SALARY
Bloom	SA_REP	10000
Fox	SA_REP	9600
Smith	SA_REP	7400
Bates	SA_REP	7300
Kumar	SA_REP	6100
Abel	SA_REP	11000
Hutton	SA_REP	8800
Taylor	SA_REP	8600
Livingston	SA_REP	8400
Johnson	SA_REP	6200

43 rows selected.

15. Menampilkan last_name dengan alias "Employee", salary dengan alias "Monthly Salary", dan commission_pct dimana commission_pct = 20%.

```
SELECT last_name "Employee", salary "Monthly
Salary", commission_pct FROM employees WHERE commission_pct = 0.2;
```

Employee	Monthly Salary	COMMISSION_PCT
Zlotkey	10500	.2
Olsen	8000	.2
Cambrault	7500	.2
Bloom	10000	.2
Fox	9600	.2
Taylor	8600	.2
Livingston	8400	.2

7 rows selected.

PRACTISE 3

1. Menampilkan sysdate (tanggal saat ini dari sistem) dengan alias "Date"

```
SELECT sysdate "Date" FROM dual;
Date
-----
08-JAN-10
```

2. Menampilkan employee_id, last_name, dan salary dimana salary mengalami kenaikan sebesar 15,5% dan hasilnya dibulatkan tanpa desimal dengan nama alias "New Salary".

```
SELECT employee_id, last_name, salary,
ROUND(salary * 1.155, 0) "New Salary" FROM employees;
```

3. Hasil Output dari soal diatas

EMPLOYEE_ID	LAST_NAME	SALARY	New Salary
190	Gates	2900	3350
191	Perkins	2500	2888
192	Bell	4000	4620
193	Everett	3900	4505
194	McCain	3200	3696
195	Jones	2800	3234
196	Walsh	3100	3581
197	Feeney	3000	3465

107 rows selected.

4. Seperti soal No. 2 & 3, tambahkan selisih dari new salary-salary yang diberi nama alias "increase"

```
SELECT employee_id, last_name, salary,
ROUND(salary * 1.155, 0) "New Salary",
ROUND(salary * 1.155, 0) - salary "Increase" FROM employees;
```

EMPLOYEE_ID	LAST_NAME	SALARY	New Salary	Increase
190	Gates	2900	3350	450
191	Perkins	2500	2888	388
192	Bell	4000	4620	620
193	Everett	3900	4505	605
194	McCain	3200	3696	496
195	Jones	2800	3234	434
196	Walsh	3100	3581	481
197	Feeney	3000	3465	465

107 rows selected.

5. Menampilkan last_name dimana Huruf pertamanya kapital dan panjang karakter last_name dimana last_name diawali huruf J, M, atau A. Urutkan berdasarkan last_name.

```
SELECT INITCAP(last_name) "Name", LENGTH(last_name) "Length"
FROM employees
WHERE last_name LIKE 'J%' OR last_name LIKE 'M%'
OR last_name LIKE 'A%' ORDER BY last_name ;
```

Name	Length
Mauris	6
Mccain	6
Mcewen	6
Mikkilineni	11
Mourgos	7

16 rows selected.

```
SELECT INITCAP(last_name) "Name", LENGTH(last_name) "Length"
FROM employees WHERE last_name LIKE '&start_letter%'
ORDER BY last_name;
```


- Menampilkan last_name dan lama bekerja (bulan) yang dihitung dengan cara waktu saat ini – hire_date beri nama dengan alias “Months_Worked”.

```
SELECT last_name, ROUND(MONTHS_BETWEEN(SYSDATE, hire_date))
MONTHS_WORKED FROM employees ORDER BY months_worked;
```

LAST_NAME	MONTHS_WORKED
Gietz	187
Baer	187
De Haan	204
Ernst	224
Hunold	240
Kochhar	244
Whalen	268
King	271

107 rows selected.

- Melakukan penggabungan dengan kalimat : <employee last name> earns <salary> monthly but wants <3 times salary>. Dimana salary dikalikan 3 dan namai dengan “Dream Salaries”

```
SELECT last_name || ' earns ' || TO_CHAR(salary, 'fm$99,999.00')
|| ' monthly but wants ' || TO_CHAR(salary * 3, 'fm$99,999.00')
|| '.' "Dream Salaries" FROM employees;
```

Dream Salaries

```
-----
Gates earns $2,900.00 monthly but wants $8,700.00.
Perkins earns $2,500.00 monthly but wants $7,500.00.
Bell earns $4,000.00 monthly but wants $12,000.00.
Everett earns $3,900.00 monthly but wants $11,700.00.
McCain earns $3,200.00 monthly but wants $9,600.00.
Jones earns $2,800.00 monthly but wants $8,400.00.
Walsh earns $3,100.00 monthly but wants $9,300.00.
Feeney earns $3,000.00 monthly but wants $9,000.00.
```

107 rows selected.

- Menampilkan last_name dan salary, panjang karakter salary = 15, bila kurang isi karakter kosong dengan simbol \$. Beri nama alias “SALARY”

```
SELECT last_name, LPAD(salary, 15, '$') SALARY FROM employees;
```

LAST_NAME	SALARY
Gates	\$\$\$\$\$\$\$\$\$\$\$\$2900
Perkins	\$\$\$\$\$\$\$\$\$\$\$\$2500
Bell	\$\$\$\$\$\$\$\$\$\$\$\$4000
Everett	\$\$\$\$\$\$\$\$\$\$\$\$3900
McCain	\$\$\$\$\$\$\$\$\$\$\$\$3200
Jones	\$\$\$\$\$\$\$\$\$\$\$\$2800
Walsh	\$\$\$\$\$\$\$\$\$\$\$\$3100
Feeney	\$\$\$\$\$\$\$\$\$\$\$\$3000

107 rows selected.

9. Tampilkan last_name, hire_date, dan hari senin setelah 6 bulan dari tanggal hire_date. Beri nama alias "REVIEW".

```
SELECT last_name, hire_date,
TO_CHAR(NEXT_DAY(ADD_MONTHS(hire_date, 6), 'MONDAY'), 'fmDay, "the"
Ddspth "of" Month, YYYY') REVIEW FROM employees;
```

LAST_NAME	HIRE_DATE
Walsh	24-APR-98
Monday, the Twenty-Sixth of October, 1998	
Feeney	23-MAY-98
Monday, the Thirtieth of November, 1998	

107 rows selected.

10. Menampilkan last_name dan hire_date serta hari dari hire_date. Urutkan berdasarkan hari tersebut.

```
SELECT last_name, hire_date, TO_CHAR(hire_date, 'DAY') DAY
FROM employees ORDER BY TO_CHAR(hire_date - 1, 'd');
```

LAST_NAME	HIRE_DATE	DAY
Bell	04-FEB-96	SUNDAY
Perkins	19-DEC-99	SUNDAY
Partners	05-JAN-97	SUNDAY
Matos	15-MAR-98	SUNDAY
Stiles	26-OCT-97	SUNDAY
Cabrio	07-FEB-99	SUNDAY
Philtanker	06-FEB-00	SUNDAY
Gee	12-DEC-99	SUNDAY

11. Menampilkan last_name dan commission_pct dengan alias "COMM", apabila memiliki komisi tampilkan besarnya, tetapi bila tidak tampilkan "No Commission".

```
SELECT last_name, NVL(TO_CHAR(commission_pct), 'No Commission')
COMM FROM employees;
```

LAST_NAME	COMM
Gates	No Commission
Perkins	No Commission
Bell	No Commission
Everett	No Commission
McCain	No Commission
Jones	No Commission
Walsh	No Commission
Feeney	No Commission

107 rows selected.

12. Tampilkan 8 karakter pertama last_name, dan salary dalam bentuk *, jumlah * menyatakan ribuan dalam \$. Beri nama alias "Employees_and_Their_Salaries".

```
SELECT rpad(last_name, 8)||' '|| rpad(' ', salary/1000+1, '*')
EMPLOYEES_AND_THEIR_SALARIES FROM employees ORDER BY salary DESC;
```

```
EMPLOYEES_AND_THEIR_SALARIES
-----
King          *****
Kochhar       *****
De Haan       *****
```

13. Menggunakan fungsi decode untuk menampilkan job_id dan grade.

```
SELECT job_id, decode(job_id,'ST_CLERK', 'E','SA_REP',
'D','IT_PROG', 'C','ST_MAN', 'B','AD_PRES', 'A','0')GRADE
FROM employees;
```

```
JOB_ID      G
-----
ST_CLERK    E
ST_CLERK    E
ST_CLERK    E
ST_MAN      B
ST_MAN      B
ST_MAN      B
ST_MAN      B
ST_MAN      B
```

107 rows selected.

14. Seperti soal No. 13, tetapi menggunakan sintaks CASE untuk proses pemilihan.

```
SELECT job_id,
CASE job_id  WHEN 'ST_CLERK' THEN 'E'
             WHEN 'SA_REP' THEN 'D'
             WHEN 'IT_PROG' THEN 'C'
             WHEN 'ST_MAN' THEN 'B'
             WHEN 'AD_PRES' THEN 'A'
ELSE '0' END GRADE
FROM employees;
```

PRACTISE 4

1. Group functions work across many rows to produce one result per group. (**True**)
Fungsi grup menghasilkan 1 hasil dari tiap grup, dimana masukannya banyak.
2. Group functions include nulls in calculations. (**False**)
3. The WHERE clause restricts rows before inclusion in a group calculation. (**True**)

4. Menampilkan salary yang terbesar, terkecil, total keseluruhan dan rata-ratanya, hasilnya dibulatkan. Dan beri nama alias-nya masing-masing

```
SELECT ROUND(MAX(salary),0) "Maximum", ROUND(MIN(salary),0)
"Minimum", ROUND(SUM(salary),0) "Sum", ROUND(AVG(salary),0)
"Average" FROM employees;
```

Maximum	Minimum	Sum	Average
24000	2100	691400	6462

5. Sama dengan No. 4 tetapi ditampilkan secara grup berdasarkan job_id.

```
SELECT job_id, ROUND(MAX(salary),0) "Maximum",
ROUND(MIN(salary),0) "Minimum", ROUND(SUM(salary),0) "Sum",
ROUND(AVG(salary),0) "Average" FROM employees GROUP BY job_id;
```

JOB_ID	Maximum	Minimum	Sum	Average
FI_MGR	12000	12000	12000	12000
PU_CLERK	3100	2500	13900	2780
SA_MAN	14000	10500	61000	12200
MK_REP	6000	6000	6000	6000
AD_PRES	24000	24000	24000	24000
SA_REP	11500	6100	250500	8350
HR_REP	6500	6500	6500	6500
ST_CLERK	3600	2100	55700	2785

19 rows selected.

6. Menampilkan jumlah orang yang memiliki pekerjaan yang sama, kelompokkan berdasarkan job_id.

```
SELECT job_id, COUNT(*) FROM employees GROUP BY job_id;
```

JOB_ID	COUNT(*)
PR_REP	1
PU_CLERK	5
PU_MAN	1
SA_MAN	5
SA_REP	30
SH_CLERK	20
ST_CLERK	20
ST_MAN	5

19 rows selected.

```
SELECT job_id, COUNT(*) FROM employees
WHERE job_id = '&job_title' GROUP BY job_id;
```

```
Enter value for job_title: SA_MAN
old 3: WHERE job_id = '&job_title'
new 3: WHERE job_id = 'SA_MAN'
```

JOB_ID	COUNT(*)
SA_MAN	5

7. Menampilkan jumlah manager

```
SELECT COUNT(DISTINCT manager_id) "Number of Managers"
FROM employees;
Number of Managers
-----
                18
```

8. Menampilkan selisih nilai terbesar dan terkecil dari salary

```
SELECT MAX(salary) - MIN(salary) DIFFERENCE FROM employees;
DIFFERENCE
-----
        21900
```

9. Menampilkan manager_id dan min(salary), dimana hanya yang memiliki manager dikelompokkan berdasarkan manager_id dan min(salary) lebih besar dari 6000 urutkan berdasarkan min(salary) descending.

```
SELECT manager_id, MIN(salary) FROM employees
WHERE manager_id IS NOT NULL GROUP BY manager_id
HAVING MIN(salary) > 6000 ORDER BY MIN(salary) DESC;
```

```
MANAGER_ID MIN(SALARY)
-----
        102          9000
        205          8300
        146          7000
        145          7000
        108          6900
        147          6200
        149          6200
        148          6100
```

8 rows selected.

10. Menampilkan total employees dan total employees yang mulai bekerja tahun 1995, 1996, 1997, dan 1998.

```
SELECT COUNT(*) total,
SUM(DECODE(TO_CHAR(hire_date, 'YYYY'), 1995, 1, 0)) "1995",
SUM(DECODE(TO_CHAR(hire_date, 'YYYY'), 1996, 1, 0)) "1996",
SUM(DECODE(TO_CHAR(hire_date, 'YYYY'), 1997, 1, 0)) "1997",
SUM(DECODE(TO_CHAR(hire_date, 'YYYY'), 1998, 1, 0)) "1998"
FROM employees;
```

```
TOTAL      1995      1996      1997      1998
-----
        107         4         10         28         23
```

11. Menampilkan job_id beserta salary-nya berdasarkan department_id dan total salary untuk tiap job_id.

```
SELECT job_id "Job",  
SUM(DECODE(department_id , 20, salary)) "Dept 20",  
SUM(DECODE(department_id , 50, salary)) "Dept 50",  
SUM(DECODE(department_id , 80, salary)) "Dept 80",  
SUM(DECODE(department_id , 90, salary)) "Dept 90",  
SUM(salary) "Total" FROM employees GROUP BY job_id;
```

Job	Dept 20	Dept 50	Dept 80	Dept 90	Total
FI_MGR					12000
PU_CLERK					13900
SA_MAN			61000		61000
MK_REP	6000				6000
AD_PRES				24000	24000
SA_REP			243500		250500
HR_REP					6500
ST_CLERK		55700			55700

19 rows selected.