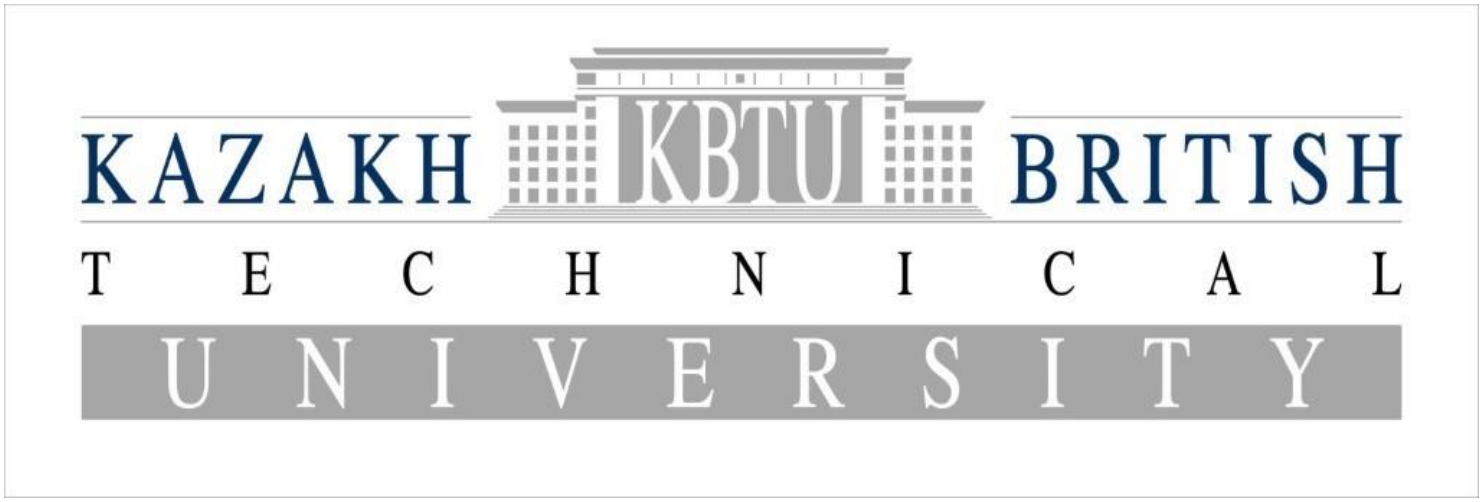

JSC "Kazakh British Technical University"
School of Mathematic and Cybernetics

Analysis of Data Bases

Laboratory Work #1

Prepared by: Maratuly Temirbolat

Almaty 2020

Main Content:

Exercise

Create Three Tables as it shown below and insert the corresponded rows

EMPLOYEES

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DAT	JOB_ID	SALARY
100	Steven	King	SKING	515.123.4567	17.06.87	AD_PRES	24000
101	Neena	Kochhar	NKOCHAR	515.123.4568	21.09.89	AD_VP	17000
102	Lex	De Haan	LDEHAAN	515.123.4569	13.03.93	AD_VP	17000
103	Alexander	Hunold	AHUNOLD	590.423.4567	03.01.90	IT_PROG	9000
104	Bruce	Ernst	BERNST	590.423.4568	21.05.91	IT_PROG	6000
107	Diana	Lorentz	DLORENTZ	590.423.5567	07.02.99	IT_PROG	4200
124	Kevin	Mourgos	KNOURGOS	650.123.5234	16.11.99	SH_MAN	5800
141	Trenna	Rajs	TRAJS	650.121.8009	17.10.95	SH_CLERK	3500
142	Curtis	Davies	CDAVIES	650.121.2994	29.01.97	SH_CLERK	3100
143	Randall	Matos	RMATOS	650.121.2874	15.03.98	SH_CLERK	2600
144	Peter	Vargas	PVARGAS	650.121.2004	09.07.98	SH_CLERK	2500
149	Eleni	Zlotkey	EZLOTKEY	011.44.1344.429018	29.01.00	SA_MAN	10500
174	Ellen	Abel	EABEL	011.44.1644.429267	11.05.96	SA_REP	11000
176	Jonathon	Taylor	JTAYLOR	011.44.1644.429265	24.03.98	SA_REP	8600
178	Kimberely	Grant	KGRANT	011.44.1644.429263	24.05.99	SA_REP	7000
200	Jennifer	Whalen	JWHALEN	515.123.4444	17.09.87	AD_ASST	4400
201	Michael	Hartstein	MHARTSTE	515.123.5555	17.02.96	MK_MAN	13000
202	Pat	Fay	PFAY	603.123.6666	17.08.97	MK_REP	6000
205	Shelley	Higgins	SHIGGINS	515.123.8080	07.06.94	AC_MGR	12000
206	William	Gietz	WGIETZ	515.123.8181	07.06.94	AC_ACCOUNT	8300

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

JOB_GRADES

GRA	LOWEST_SAL	HIGHEST_SAL
A	1000	2999
B	3000	5999
C	6000	9999
D	10000	14999
E	15000	24999
F	25000	40000

Solution:

For the first Table:

```
create table EMPLOYEES(     /* The line shows the request to create a table with the variables for each columns*/
    EMPLOYEE_ID integer,    /* Each colum has own name and data type */
    FIRST_NAME varchar(20),
    LAST_NAME varchar(20),
    EMAIL varchar(40),
    PHONE_NUMBER varchar(30),
    HIRE_DAT date,
    JOB_ID varchar(20),
    SALARY integer
);
insert into EMPLOYEES values ('206','William','Gietz','WGIETZ','515.123.8181', '07.06.94','AC_ACCOUNT','8300');
/* The request "insert" means to add the new row with to the table EMPLOYEE with parameters in order as we did duriing the creation*/
/* It is written only one time because I was correcting and rechanging the data during the inserting*/
select * from EMPLOYEES; /* The request "select" with the star shows all the columns from the EMPLOYEE table that exist */
```

Result:

	employee_id	first_name	last_name	email	phone_number	hire_dat	job_id	salary
1	100	Assanali	Moldash	SKING	515.123.4567	2001-08-18	AD_PRES	24000
2	101	Neena	Kochhar	NKOCHAR	515.123.4568	1989-09-21	AD_VP	17000
3	102	Lex	De Haan	LDEHAAN	515.123.4569	1993-03-13	AD_VP	17000
4	103	Alexander	Hunold	AHUNOLD	515.123.4567	1990-01-03	IT_PROG	9000
5	104	Bruce	Ernst	BERNST	515.123.4568	1991-05-21	IT_PROG	6000
6	107	Diana	Lorentz	DLORENTZ	515.123.4567	1999-02-07	IT_PROG	4200
7	124	Kevin	Mourgos	KNOURGOS	650.123.5234	1999-11-16	SH_MAN	5800
8	141	Trenna	Rajs	TRAJS	650.121.8009	1995-10-17	SH_CLERK	3500
9	142	Curtis	Davies	CDAVIES	650.121.2994	1997-01-29	SH_CLERK	3100
10	143	Randall	Matos	RMATOS	650.121.2874	1998-03-15	SH_CLERK	2600
11	144	Peter	Vargas	PVARGAS	650.121.2004	1998-07-09	SH_CLERK	2500
12	149	Eleni	Zlotkey	EZLOTKEY	011.44.1344.429018	2000-01-29	SA_MAN	10500
13	174	Ellen	Abel	EABEL	011.44.1644.429267	1996-05-11	SA_REP	11000
14	176	Jonathon	Taylor	JTAYLOR	011.44.1644.429265	1998-03-24	SA_REP	8600
15	178	Kimberely	Grant	KGRANT	011.44.1644.429263	1999-05-24	SA_REP	7000
16	200	Jennifer	Whalen	JWHALEN	515.123.4444	1987-09-17	AD_ASST	4400
17	201	Michael	Hartstein	MHARTSTE	515.123.5555	1996-02-17	MK_MAN	13000
18	202	Pat	Fay	PFAY	603.123.6666	1997-08-17	MK_REP	6000
19	205	Shelley	Higgins	SHIGGINS	515.123.8080	1994-06-07	AC_MGR	12000
20	206	William	Gietz	WGIETZ	515.123.8181	1994-06-07	AC_ACCOUNT	8300

For the Second:

```
create table departments( /* The line creates table with the columns that have data types correspond to them*/
    department_id integer,
    department_name varchar(25), /*The length of the word can be no more the 25 */
    manager_id integer,
    location_id integer
);

insert into departments values ('10','Administration','200','1700'); /* Lines from the 24 to 31 insert (add) new rows with the data to the table*/
insert into departments values ('20','Marketing','201','1800');
insert into departments values ('50','Shipping','124','1500');
insert into departments values ('60','IT','103','1400');
insert into departments values ('80','Sales','149','2500');
insert into departments values ('90','Executive','100','1700');
insert into departments values ('110','Accounting','205','1700');
insert into departments values ('190','Contracting',NULL,'1700');

select * from departments; /* Here we shows all the columns with the filled information from the table "departments"*/
```

The Resulted Table:

	department_id	department_name	manager_id	location_id
1	10	Administration	200	1700
2	20	Marketing	201	1800
3	50	Shipping	124	1500
4	60	IT	103	1400
5	80	Sales	149	2500
6	90	Executive	100	1700
7	110	Accounting	205	1700
8	190	Contracting	<null>	1700

For the Third Table:

```
create table job_grades(   /* The line creates a table with the columns that have data types*/
    gra varchar(1),         /*The length of the word is only 1*/
    lowest_sal integer,
    highest_sal integer
);

insert into job_grades values ('A','1000','2999');  /* Lines from the 41 to 46 insert new rows with the data to the table*/
insert into job_grades values ('B','3000','5999');
insert into job_grades values ('C','6000','9999');
insert into job_grades values ('D','10000','14999');
insert into job_grades values ('E','15000','24999');
insert into job_grades values ('F','25000','40000');

select * from job_grades; /*Request to show all the columns from the "job_grades"*/
```

The resulted table is:

	gra	lowest_sal	highest_sal
1	A	1000	2999
2	B	3000	5999
3	C	6000	9999
4	D	10000	14999
5	E	15000	24999
6	F	25000	40000

Exercises:

1. The Human Resources (HR) department needs data including id, first name, last name, hiring date and salaries of all employees

```
select employee_id, first_name,last_name, hire_dat,salary from EMPLOYEES; /* show the data including id, first name, last name, hiring
date and salaries of all employees*/
```

	employee_id	first_name	last_name	hire_dat	salary
1	100	Assanali	Moldash	2001-08-18	24000
2	101	Neena	Kochhar	1989-09-21	17000
3	102	Lex	De Haan	1993-03-13	17000
4	103	Alexander	Hunold	1990-01-03	9000
5	104	Bruce	Ernst	1991-05-21	6000
6	107	Diana	Lorentz	1999-02-07	4200
7	124	Kevin	Mourgos	1999-11-16	5800
8	141	Trenna	Rajs	1995-10-17	3500
9	142	Curtis	Davies	1997-01-29	3100
10	143	Randall	Matos	1998-03-15	2600
11	144	Peter	Vargas	1998-07-09	2500
12	149	Eleni	Zlotkey	2000-01-29	10500
13	174	Ellen	Abel	1996-05-11	11000
14	176	Jonathon	Taylor	1998-03-24	8600
15	178	Kimberely	Grant	1999-05-24	7000
16	200	Jennifer	Whalen	1987-09-17	4400
17	201	Michael	Hartstein	1996-02-17	13000
18	202	Pat	Fay	1997-08-17	6000
19	205	Shelley	Higgins	1994-06-07	12000
20	206	William	Gietz	1994-06-07	8300

2. Write a query to display id, first names, last names, email, annual salaries of all empl

```
select employee_id,first_name,last_name,email,12*salary from EMPLOYEES; /*Write a query to display id, first names, last names, email,
                                                                          annual salaries of all employees.*/
```

	employee_id	first_name	last_name	email	?column?
1	100	Assanali	Moldash	SKING	288000
2	101	Neena	Kochhar	NKOCHAR	204000
3	102	Lex	De Haan	LDEHAAN	204000
4	103	Alexander	Hunold	AHUNOLD	108000
5	104	Bruce	Ernst	BERNST	72000
6	107	Diana	Lorentz	DLORENTZ	50400
7	124	Kevin	Mourgos	KNOURGOS	69600
8	141	Trenna	Rajs	TRAJS	42000
9	142	Curtis	Davies	CDAVIES	37200
10	143	Randall	Matos	RMATOS	31200
11	144	Peter	Vargas	PVARGAS	30000
12	149	Eleni	Zlotkey	EZLOTKEY	126000
13	174	Ellen	Abel	EABEL	132000
14	176	Jonathon	Taylor	JTAYLOR	103200
15	178	Kimberely	Grant	KGRANT	84000
16	200	Jennifer	Whalen	JWHALEN	52800
17	201	Michael	Hartstein	MHARTSTE	156000
18	202	Pat	Fay	PFAY	72000
19	205	Shelley	Higgins	SHIGGINS	144000
20	206	William	Gietz	WGIETZ	99600

3. The Human Resources (HR) department requests data for all unique jobs from the EMPLOYEES table. Job IDs should not be repeated in the output.

```
22 select DISTINCT job_id from EMPLOYEES; /*The Human Resources (HR) department requests data for all unique jobs from the
23 EMPLOYEES table. Job IDs should not be repeated in the output*/
```

	job_id
1	AD_VP
2	AC_MGR
3	AD_PRES
4	SH_MAN
5	MK_MAN
6	AD_ASST
7	IT_PROG
8	SA_REP
9	MK_REP
10	SA_MAN
11	SH_CLERK
12	AC_ACCOUNT

4. Due to funding problems, the HR department needs a report that provides all the information about the programmers whose salaries are over 5000

```
select * from EMPLOYEES where job_id='IT_PROG' AND salary >5000; /*THE report that provides all the information
about the programmers whose salaries are over 5000.*/
```

	employee_id	first_name	last_name	email	phone_number	hire_dat	job_id	salary
1	103	Alexander	Hunold	AHUNOLD	515.123.4567	1990-01-03	IT_PROG	9000
2	104	Bruce	Ernst	BERNST	515.123.4568	1991-05-21	IT_PROG	6000

5. Generate a report to display the id, last name, first name, and job title of all employees whose salaries range from 4000 to 7000.

```
select employee_id, last_name,first_name,job_id from EMPLOYEES where salary>=4000 AND salary<=7000; /* Display the id, last name, first name, and job title
of all employees whose salaries range from 4000 to 7000.*/
```

	employee_id	last_name	first_name	job_id
1	104	Ernst	Bruce	IT_PROG
2	107	Lorentz	Diana	IT_PROG
3	124	Mourgos	Kevin	SH_MAN
4	178	Grant	Kimberely	SA_REP
5	200	Whalen	Jennifer	AD_ASST
6	202	Fay	Pat	MK_REP

6. The HR department needs data on high-paid and low-paid employees. Write a query to display the last names, first names, and salaries of all employees whose salaries are outside the range from 3000 to 9000.

```
select last_name,first_name,salary from EMPLOYEES where salary<3000 OR salary >9000;
/*display the last names, first names, and salaries of all employees whose salaries are outside the range from 3000 to 9000.*/
```

	last_name	first_name	salary
1	Moldash	Assanali	24000
2	Kochhar	Neena	17000
3	De Haan	Lex	17000
4	Matos	Randall	2600
5	Vargas	Peter	2500
6	Zlotkey	Eleni	10500
7	Abel	Ellen	11000
8	Hartstein	Michael	13000
9	Higgins	Shelley	12000

7. Write a query to display id, last names, first names, annual salaries of those employees whose salaries are below 10000.

```
select employee_id,last_name,first_name, 12*employees.salary from EMPLOYEES where salary<10000;
/*Display id, last names, first names, annual salaries of those employees whose salaries are below 10000.*/
```

	employee_id	last_name	first_name	?column?
1	103	Hunold	Alexander	108000
2	104	Ernst	Bruce	72000
3	107	Lorentz	Diana	50400
4	124	Mourgos	Kevin	69600
5	141	Rajs	Trenna	42000
6	142	Davies	Curtis	37200
7	143	Matos	Randall	31200
8	144	Vargas	Peter	30000
9	176	Taylor	Jonathon	103200
10	178	Grant	Kimberely	84000
11	200	Whalen	Jennifer	52800
12	202	Fay	Pat	72000
13	206	Gietz	William	99600

8. Write a query to display id, last names, salaries of those employees whose salaries are in the range from 4000 to 7000 using the BETWEEN ... AND command and explain the difference between this task and task #5.

The Difference between these two tasks is that in 5-th we used typical conditions (>= ,<=) with the binary statement as (AND) to determine the range using mathematical point of view, but in 8-th exercise we used “Between… AND… ” command to determine the range having the correspondent function. This is also suitable for texts, dates.

```
37 select employee_id,last_name,salary from EMPLOYEES where SALARY BETWEEN 4000 AND 7000;/*Display id, last names, salaries of those employees
38 whose salaries are in the range from 4000 to 7000 using the BETWEEN ... AND command */
```

	employee_id	last_name	salary
1	104	Ernst	6000
2	107	Lorentz	4200
3	124	Mourgos	5800
4	178	Grant	7000
5	200	Whalen	4400
6	202	Fay	6000

9. Write a query to display id, last names, salaries, job title from the list of id «144, 102, 200, 205».

```
40 select employee_id, last_name,salary,job_id from EMPLOYEES where EMPLOYEE_ID IN (144,102,200,205);
41 /*Write a query to display id, last names, salaries, job title from the list of id «144, 102, 200, 205»*/
42 /*Used "IN " because we have several about the same conditions in order to make the raw smaller*/
```

	employee_id	last_name	salary	job_id
1	102	De Haan	17000	AD_VP
2	144	Vargas	2500	SH_CLERK
3	200	Whalen	4400	AD_ASST
4	205	Higgins	12000	AC_MGR

10.Write a query to display id, last names, salaries, job title not from the list of id «144, 102, 200, 205».

```
44 select employee_id,last_name,salary,job_id from EMPLOYEES where EMPLOYEE_ID NOT IN (144,102,200,205);
45 /*Display id, last names, salaries, job title not from the list of id «144, 102, 200, 205».*/
```

	employee_id	last_name	salary	job_id
1	100	Moldash	24000	AD_PRES
2	101	Kochhar	17000	AD_VP
3	103	Hunold	9000	IT_PROG
4	104	Ernst	6000	IT_PROG
5	107	Lorentz	4200	IT_PROG
6	124	Mourgos	5800	SH_MAN
7	141	Rajs	3500	SH_CLERK
8	142	Davies	3100	SH_CLERK
9	143	Matos	2600	SH_CLERK
10	149	Zlotkey	10500	SA_MAN
11	174	Abel	11000	SA_REP
12	176	Taylor	8600	SA_REP
13	178	Grant	7000	SA_REP
14	201	Hartstein	13000	MK_MAN
15	202	Fay	6000	MK_REP
16	206	Gietz	8300	AC_ACCOUNT

11. Write a query to display id, last names, first names, salaries of those employees whose names begin with the letter D.

```
47  select employee_id, last_name,first_name,salary from EMPLOYEES where first_name like 'D%';
48  /* Display id, last names, first names, salaries of those employees whose names begin with the letter D.*/
```

	employee_id	last_name	first_name	salary
1	107	Lorentz	Diana	4200

12. Write a query to display all the names of employees in which the third letter is A.
For 'A' (Capital A)

```
select first_name from EMPLOYEES where first_name like '__A%'; /*Here __ means only two any letters and then A */
/*Display all the names of employees in which the third letter is A.*/
```

No correspondent data in the table

For 'a' (small a):

```
50  select first_name from EMPLOYEES where first_name like '__a%'; /*Here __ means only two any letters and then a */
51  /*Display all the names of employees in which the third letter is a.*/
```

	first_name
1	Diana

13. . Write a query to display id, last names, names, email, salaries of those employees whose names end with a letter N.
For Capital 'N'. We get Empty Table

```
53  select employee_id,last_name,first_name,email,salary from EMPLOYEES where first_name like '%N';
54  /* % means that instead of it can be nothing, one symbol or several and then N in the end*/
55  /*Display id, last names, names, email, salaries of those employees whose names end with a letter N.*/
```

employee_id	last_name	first_name	email	salary

For small 'n'.

```
53  select employee_id,last_name,first_name,email,salary from EMPLOYEES where first_name like '%n';
54  /* % means that instead of it can be nothing, one symbol or several and then n in the end*/
55  /*Display id, last names, names, email, salaries of those employees whose names end with a letter n.*/
```

	employee_id	last_name	first_name	email	salary
1	124	Mourgos	Kevin	KNOURGOS	5800
2	174	Abel	Ellen	EABEL	11000
3	176	Taylor	Jonathon	JTAYLOR	8600

14. Write a query to display id, last names, email, salaries of all employees, sorting their salaries in ascending order

```
57  select employee_id, last_name,email,salary from EMPLOYEES ORDER BY salary ASC;
58  /*Display id, last names, email, salaries of all employees, sorting their salaries in ascending order*/
```

	employee_id	last_name	email	salary
1	144	Vargas	PVARGAS	2500
2	143	Matos	RMATOS	2600
3	142	Davies	CDAVIES	3100
4	141	Rajs	TRAJS	3500
5	107	Lorentz	DLORENTZ	4200
6	200	Whalen	JWHALEN	4400
7	124	Mourgos	KNOURGOS	5800
8	104	Ernst	BERNST	6000
9	202	Fay	PFAY	6000
10	178	Grant	KGRANT	7000
11	206	Gietz	WGIETZ	8300
12	176	Taylor	JTAYLOR	8600
13	103	Hunold	AHUNOLD	9000
14	149	Zlotkey	EZLOTKEY	10500
15	174	Abel	EABEL	11000
16	205	Higgins	SHIGGINS	12000
17	201	Hartstein	MHARTSTE	13000
18	102	De Haan	LDEHAAN	17000
19	101	Kochhar	NKOCHAR	17000
20	100	Moldash	SKING	24000

15. Write a query to display id, last names, names, salaries of all employees, sorting their id in descending order

```
60 ✔ select employee_id, last_name,first_name,salary from EMPLOYEES ORDER BY EMPLOYEE_ID DESC ;
61   /*Display id, last names, names, salaries of all employees, sorting their id in descending order*/
```

	employee_id	last_name	first_name	salary
1	206	Gietz	William	8300
2	205	Higgins	Shelley	12000
3	202	Fay	Pat	6000
4	201	Hartstein	Michael	13000
5	200	Whalen	Jennifer	4400
6	178	Grant	Kimberely	7000
7	176	Taylor	Jonathon	8600
8	174	Abel	Ellen	11000
9	149	Zlotkey	Eleni	10500
10	144	Vargas	Peter	2500
11	143	Matos	Randall	2600
12	142	Davies	Curtis	3100
13	141	Rajs	Trenna	3500
14	124	Mourgos	Kevin	5800
15	107	Lorentz	Diana	4200
16	104	Ernst	Bruce	6000
17	103	Hunold	Alexander	9000
18	102	De Haan	Lex	17000
19	101	Kochhar	Neena	17000
20	100	Moldash	Assanali	24000

16. Write a query to display the average, maximum, minimum and the sum of all programmers' salaries.

```
63 ✔ select AVG(salary) as average_salary,MAX(salary) as maximum_salary, MIN(salary) as minimum_salary, SUM(salary) as sum_salary
64   From EMPLOYEES where JOB_ID = 'IT_PROG';
65   /*Display the average, maximum, minimum and the sum of all programmers' salaries.*/
```

	average_salary	maximum_salary	minimum_salary	sum_salary
1	6400	9000	4200	19200

17. Write a query to display the number of programmers.

```
67 ✔ select COUNT(job_id) as programmer_number from EMPLOYEES where JOB_ID = 'IT_PROG';
68   /*Display the number of programmers. */
```

	programmer_number
1	3

18. Write a query to display the number of unique professions.

```
70  select  COUNT(DISTINCT job_id) as number_unique_jobs from EMPLOYEES;
71  /*Display the number of unique professions.*/
```

	number_unique_jobs
1	12

19. Sum the salaries in the EMPLOYEES table for each job title.

```
73  select SUM(salary),JOB_ID from EMPLOYEES group by  JOB_ID;
74  /*Show the  sum of the salaries in the EMPLOYEES table for each job title. */
```

	sum	job_id
1	34000	AD_VP
2	12000	AC_MGR
3	24000	AD_PRES
4	5800	SH_MAN
5	13000	MK_MAN
6	4400	AD_ASST
7	19200	IT_PROG
8	26600	SA_REP
9	6000	MK_REP
10	10500	SA_MAN
11	11700	SH_CLERK
12	8300	AC_ACCOUNT

20. Find the average salaries in the EMPLOYEES table for each job title.

```
76  select AVG(salary), JOB_ID from EMPLOYEES group by JOB_ID;
77  /*Average salaries in the EMPLOYEES table for each job title. */
```

	avg	job_id
1	17000	AD_VP
2	12000	AC_MGR
3	24000	AD_PRES
4	5800	SH_MAN
5	13000	MK_MAN
6	4400	AD_ASST
7	6400	IT_PROG
8	8866.6666666666666667	SA_REP
9	6000	MK_REP
10	10500	SA_MAN
11	2925	SH_CLERK
12	8300	AC_ACCOUNT

21. Find the maximum salaries in the EMPLOYEES table for each job title that exceed 10,000 and sort them in descending order.

```
79  select MAX(salary), JOB_ID from EMPLOYEES group by JOB_ID HAVING MAX(salary) >10000 order by  MAX(SALARY) DESC ;
80  /*Maximum salaries in the EMPLOYEES table for each job title that exceed 10,000 and
81  sort them in descending order. */
```

	max	job_id
1	24000	AD_PRES
2	17000	AD_VP
3	13000	MK_MAN
4	12000	AC_MGR
5	11000	SA_REP
6	10500	SA_MAN

22. Find the maximum average salary for each job title.

```
83  select MAX(average_salary) as maximum_average_salary from (
84                             select AVG(salary) average_salary
85                             from EMPLOYEES
86                             GROUP BY JOB_ID
87                                 )a ;
88  /*. Maximum average salary for each job title.*/
```

	maximum_average_salary
1	24000