

UNIVERSIDAD AUTÓNOMA DE ZACATECAS
INGENIERÍA DE SOFTWARE
SISTEMAS DE BASE DE DATOS II

Ejercicio del Capítulo 3

A) Elabore las siguientes consultas sobre la base de datos HR:

- 1. Mencione la diferencia que hay entre las funciones SINGLE-ROW y MULTIPLE-ROW.**
 - i. Las funciones single-row toman como argumento una sola línea y devuelven un solo valor, en cambio las funciones multiple-row obtienen varias filas de entrada y devuelven un resultado de salida.
- 2. Mencione 5 características de una función SINGLE-ROW.**
 - i. Manipular elementos de datos.
 - ii. Acepta argumentos y devuelve un valor.
 - iii. Actuar en cada fila que se devuelve.
 - iv. Devuelve un resultado por fila.
 - v. Puede modificar el tipo de datos.
- 3. Las funciones SINGLE-ROW se pueden clasificar en:**
 - i. Funciones de carácter.
 - ii. Funciones generales.
 - iii. Funciones numéricas.
 - iv. Funciones de conversión.
 - v. Funciones de fechas.
- 4. Escriba una sentencia SELECT que muestre el siguiente resultado:**

	A2	EMPLOYEE DETAILS
1		The job id for ABEL is sa_rep
2		The job id for DAVIES is st_clerk
3		The job id for DE HAAN is ad_vp
4		The job id for ERNST is it_prog
5		The job id for FAY is mk_rep
6		The job id for GIETZ is ac_account

...

- i. SELECT 'The job if from '||LAST_name||' is '||job_id AS "EMPLOYEE DETAILS" FROM EMPLOYEES;

5. Explique la diferencia que hay entre las siguientes consultas:

```
SELECT employee_id, last_name, department_id
FROM employees
WHERE last_name = 'higgins';
```

0 rows selected

```
SELECT employee_id, last_name, department_id
FROM employees
WHERE LOWER(last_name) = 'higgins';
```

	A2	EMPLOYEE_ID	A2	LAST_NAME	A2	DEPARTMENT_ID
1		205		Higgins		110

- i. La primera consulta verifica que el apellido sea totalmente igual a la cadena que está totalmente en minúsculas, esto incluye verificar mayúsculas y minúsculas, en la segunda consulta hace que el apellido sea en minúsculas para compararlo con la cadena que esta en minúsculas.

6. Elabore una sentencia SELECT que muestre el siguiente resultado (Solo muestra a los empleados que su last_name termina con la letra "n"):

	A2	EMPLOYEE_ID	A2	NAME	A2	LENGTH(LAST_NAME)	A2	Contains 'a'?
1		102		LexDe Haan		7		5
2		200		JenniferWhalen		6		3
3		201		MichaelHartstein		9		2

- i. Select employee_id, first_name || ' ' || last_name as "NAME", LENGTH(LAST_NAME), instr(last_name,'a') as "Contains 'a'?" from Employees where last_name like '%n';

7. Explique el comportamiento de las siguientes sentencias:

Query 1: `select initcap(21/7) from dual`

Query 2: `select initcap(SYSDATE) from dual`

Query 3: `select initcap('init cap or init_cap or init%cap') from dual`

- i. La primera consulta hace una división haciendo uso de la tabla dual la cual contiene una columna y un registro y con esto lograr que regrese solo el resultado querido.
- ii. La segunda consulta regresa la fecha actual en la que se ejecuta, también lo hace usando la tabla dual.
- iii. En las dos primeras consultas la función `initcap` no hace nada y no es necesario ponerla. En la tercera consulta hace que la cadena tenga mayúsculas en cada palabra que es separada por un espacio, guion bajo o un signo de %.

8. Elabore una sentencia SELECT que muestre un listado de países cuyo nombre tenga una longitud mínima de 10 caracteres.

- i. `select * from countries where length(country_name) >= 10;`

9. Explique el comportamiento de las siguientes sentencias:

Query 1: `select lpad(1000+200.55,14,'*') from dual`

Query 2: `select rpad(1000+200.55,14,'*') from dual`

Query 3: `select lpad(SYSDATE,14,'$#') from dual`

Query 4: `select rpad(SYSDATE,4,'$#') from dual`

- i. La primera consulta coloca 7 asteriscos antes del número resultante de la suma el cual son 7 caracteres y rellena con otros 7 de asteriscos para completar los 14 caracteres.
- ii. La segunda coloca los 7 asteriscos después del número resultante.
- iii. La tercera consulta rellena con los 2 caracteres dados hasta que sean 14 caracteres lo que resulte.
- iv. El ultimo solo obtiene los primeros 4 caracteres de la fecha resultante

10. Explique el comportamiento de las siguientes sentencias:

Query 1: `select trim(trailing 'e' from 1+2.14||' is pie') from dual`

Query 2: `select trim(both '*' from '*****Hidden*****') from dual`

Query 3: `select trim(1 from sysdate) from dual`

- i. La primera consulta quita todas las e de la cadena dada.
- ii. La segunda consulta elimina todos los asteriscos de la cadena.
- iii. El tercero elimina todos los 1 de la fecha resultante

11. Explique el comportamiento de las siguientes sentencias:

Query 3: `select replace('1#3#5#7#9#','#','>') from dual`

Query 4: `select replace('1#3#5#7#9#','#') from dual`

- i. El primero reemplaza los símbolos # por ">" y el segundo al no tener con que reemplazar los símbolos los elimina.

12. Explique el comportamiento de las siguientes sentencias:

Query 1: `select round(1601.916718,1) from dual`

Query 2: `select round(1601.916718,2) from dual`

Query 3: `select round(1601.916718,-3) from dual`

Query 4: `select round(1601.916718) from dual`

- i. Redondea los decimales dependiendo de los dígitos que se quieren usar. En caso de colocar un numero negativo la función dará error. Si no se le especifican los dígitos que se usaran, se redondea a un número entero.

13. Explique el comportamiento de las siguientes sentencias:

Query 1: `select trunc(1601.916718,1) from dual`

Query 2: `select trunc(1601.916718,2) from dual`

Query 3: `select trunc(1601.916718,-3) from dual`

Query 4: `select trunc(1601.916718) from dual`

- i. La función trunc lo que hará es solo mostrar los dígitos que se le especifiquen, si se le especifica un numero negativo tomara los números enteros y los hará 0, si no se le especifica, solo eliminara los dígitos después del punto, todo esto sin redondear.

14. Explique el comportamiento de las siguientes sentencias:

Query 1: `select mod(6,2) from dual`

Query 2: `select mod(5,3) from dual`

Query 3: `select mod(7,35) from dual`

Query 4: `select mod(5.2,3) from dual`

- i. Hace la operación de modulo y obtiene el rsuido.

15. Interprete el formato RR y el formato YY para cada ejemplo (llenar cada espacio en blanco con el año en 4 dígitos (ej. 1995) según corresponda).

Current year	Given date	Interpreted (RR)	Interpreted (YY)
1994	25-oct-95	1995	1995
1994	27-oct-17	2017	1917
2001	27-oct-17	2017	2017
2048	27-oct-52	1952	2052
2051	27-oct-47	2047	2047

16. Elabore una sentencia SELECT que muestre el siguiente resultado.

	EMPLOYEE_ID	HIRE_DATE	TENURE	REVIEW	NEXT_DA...	LAST_DAY...
1	202	17-AUG-97	141.79757989...	17-FEB-98	22-AUG-97	31-AUG-97
2	107	07-FEB-99	124.12016054...	07-AUG-99	12-FEB-99	28-FEB-99
3	124	16-NOV-99	114.82983796...	16-MAY-00	19-NOV-99	30-NOV-99
4	142	29-JAN-97	148.41048312...	29-JUL-97	31-JAN-97	31-JAN-97
5	143	15-MAR-98	134.86209602...	15-SEP-98	20-MAR-98	31-MAR-98
6	144	09-JUL-98	131.05564441...	09-JAN-99	10-JUL-98	31-JUL-98
7	149	29-JAN-00	112.41048312...	29-JUL-00	04-FEB-00	31-JAN-00
8	176	24-MAR-98	134.57177344...	24-SEP-98	27-MAR-98	31-MAR-98
9	178	24-MAY-99	120.57177344...	24-NOV-99	28-MAY-99	31-MAY-99

- i. Select employee_id, hire_date, MONTHS_BETWEEN(hire_date,sysdate) TENURE, (add_months(hire_date, 6)) REVIEW, next_day(hire_date, 1), last_day(hire_date) from Employees;

17. Elabore una sentencia SELECT que muestre el siguiente resultado.

	EMPLOYEE_ID	HIRE_DATE	ROUND(HIRE_DATE,'MONTH')	TRUNC(HIRE_DATE,'MONTH')
1	202	17-AUG-97	01-SEP-97	01-AUG-97
2	142	29-JAN-97	01-FEB-97	01-JAN-97

- i. Select employee_id, hire_date, round(hire_date, 'MONTH'), trunc(hire_date, 'MONTH') from employees;

B) Elabore 2 consultas sobre la base de datos “HOTEL” para cada una de las siguientes opciones:

1. Lower

- i. Select lower(nombre||' '||apellido_paterno||' '||apellido_materno) as "Nombre" from Cliente_jmjp;
- ii. Select * from Hoteles_jmjp where lower(colonia) = 'h_col_4';

2. Upper

- i. Select * from Tipo_Habitacion_jmjp where descripcion = upper('habitacion para un huesped');
- ii. Select nombre, telefono from Agencia_jmjp where nombre = upper('AgEnCiA 1');

3. Initcap
 - i. Select initcap(nombre||' '||apellido_paterno||' '||apellido_materno) as "Nombre" from Cliente_jmjp;
 - ii. Select ID_HABITACION,initcap(NOMBRE) nombre,ID_TIPO from Habitaciones_jmjp;
4. Concat
 - i. select concat(concat(concat(concat(nombre,' '),apellido_paterno),' '),apellido_materno) as "Nombre" from Cliente_jmjp;
 - ii. Select concat(concat(concat('La habitacion con el id ',ID_HABITACION),' tiene el nombre '),initcap(nombre)) as Habitacion from Habitaciones_jmjp;
5. Substr
 - i. Select substr(descripcion,12,25) from tipo_habitacion_jmjp;
 - ii. Select substr(nombre,1,3) nombre, substr(apellido_paterno,1,3) from Empleado_jmjp;
6. Length
 - i. Select nombre, length(nombre) from Cliente_jmjp;
 - ii. Select nombre, length(nombre), apellido_paterno, length(apellido_paterno), apellido_materno, length(apellido_materno) from Empleado_jmjp;
7. Instr
 - i. Select nombre, instr(nombre,'E') from Hoteles_jmjp;
 - ii. Select nombre, instr(apellido_materno,'A'), apellido_paterno, instr(apellido_paterno,'A') from Cliente_jmjp;
8. Lpad | Rpad
 - i. Select
lpad(rpad(nombre,length(nombre)+1,'⊙'),length(nombre)+2,'⊙')
"⊙ Nombre ⊙" from Cliente_jmjp;
 - ii. Select lpad(rpad(nombre,length(nombre)+3,'*'),length(nombre)*2,'*')
nombre from Hoteles_jmjp;
9. Trim
 - i. Select trim(trailing 'S' from descripcion) Descripcion from Categorias_jmjp;
 - ii. Select trim(both 'Z' from (nombre||' '||apellido_paterno||' '||apellido_materno)) from Empleado_jmjp;

10. Replace

- i. Select id_cliente,replace(telefono,'836','***) numero from Cliente_jmjp;
- ii. Select id_hotel,replace(fecha_construccion,'/','-') fecha_construccion from Hoteles_jmjp;

11. Round

- i. Select id_reserva,fecha_inicio,fecha_fin,round(precio,-1) from Reserva_jmjp;
- ii. Select id_reserva,fecha_inicio,fecha_fin,round(precio) from Reserva_jmjp;

12. Trunc

- i. Select id_reserva,fecha_inicio,fecha_fin,trunc(precio,-1) from Reserva_jmjp;
- ii. Select id_reserva,fecha_inicio,fecha_fin,trunc(precio,-3) from Reserva_jmjp;

13. Mod

- i. Select id_reserva,fecha_inicio,fecha_fin,precio-mod(precio,15) as "Precio" from Reserva_jmjp;
- ii. Select id_reserva,fecha_inicio,fecha_fin,precio-mod(precio,150) as "Precio" from Reserva_jmjp;

14. Round para fechas

- i. Select id_hotel,round(fecha_construccion,'MONTH') fecha,nombre,telefono from Hoteles_jmjp;
- ii. Select id_reserva,round(fecha_inicio,'MONTH') fecha, precio from Reserva_jmjp;

15. Trunc para fechas

- i. Select id_hotel,trunc(fecha_construccion,'MONTH') fecha,nombre,telefono from Hoteles_jmjp;
- ii. Select id_reserva,trunc(fecha_inicio,'MONTH') fecha, precio from Reserva_jmjp;

16. Operadores aritméticos para fechas

- i. Select id_hotel, nombre, fecha_construccion from Hoteles_jmjp where fecha_construccion > '11/11/12';
- ii. Select id_reserva,fecha_inicio,fecha_fin,precio from Reserva_jmjp where fecha_fin-fecha_inicio > 3;

17. Months_Between

- i. Select id_hotel, nombre, fecha_construccion, months_between(fecha_construccion,sysdate) as meses from Hoteles_jmjp;

- ii. `Select id_reserva, fecha_inicio, fecha_fin, precio, months_between(fecha_inicio, add_months(fecha_fin, 16)) as fecha_inicio from Reserva_jmjp;`

18. Add_months

- i. `Select id_hotel, nombre, add_months(fecha_construccion, 12) from Hoteles_jmjp;`
- ii. `Select id_reserva, add_months(fecha_inicio, 1) fecha_inicio, add_months(fecha_fin, 2) fecha_fin from Reserva_jmjp;`

19. Next_day

- i. `Select id_hotel, nombre, next_day(fecha_construccion, 2) from Hoteles_jmjp;`
- ii. `Select id_reserva, next_day(fecha_inicio, 1) fecha_inicio, next_day(fecha_fin, 5) fecha_fin from Reserva_jmjp;`

20. Last_day

- i. `Select id_hotel, nombre, last_day(fecha_construccion) from Hoteles_jmjp;`
- ii. `Select id_reserva, last_day(fecha_inicio) fecha_inicio, last_day(fecha_fin) fecha_fin from Reserva_jmjp;`