**Ejercicios Insert, Update, Delete**

1. **INSERT**
   1. Insertar un nuevo departamento con código 55, nombre RECURSOS y localidad MADRID.

**INSERT INTO DEPART**

**SELECT 55, 'RECURSOS', 'MADRID'**

**FROM DUAL;**

**INSERT INTO DEPART VALUES (55, 'RECURSOS', 'MADRID');**

* 1. Insertar un empleado en el departamento 55 con número de empleado 8043, apellido ESCUDERO, salario 1230€, oficio EMPLEADO y fecha de alta la del sistema.

**INSERT INTO EMPLE**

**SELECT 8043, 'ESCUDERO', 'EMPLEADO', NULL, SYSDATE, 1230, NULL, 55**

**FROM DUAL;**

* 1. Insertar un empleado con número 8001, apellido ROJO, fecha de alta ayer y el resto de datos los del empleado 7654.

**INSERT INTO EMPLE**

**SELECT 8001, 'ROJO', OFICIO, RESPONSABLE, SYSDATE-1, SALARIO, COMISION, DEPT\_NO**

**FROM EMPLE**

**WHERE EMP\_NO = 7654;**

* 1. Insertar un empleado con el número 9050, apellido el tuyo, oficio Becario, responsable el número de empleado de SALA, fecha de alta el lunes de la semana que viene, salario la mitad del de SALA, comisión nula y departamento el de SALA

**INSERT INTO EMPLE**

**SELECT 9050, 'BATISTA', 'BECARIO', RESPONSABLE, SYSDATE+5, SALARIO/2, NULL, DEPT\_NO**

**FROM EMPLE**

**WHERE APELLIDO LIKE 'SALA';**

* 1. Insertar un nuevo departamento con código 65, nombre ‘I+D’ y localidad la del departamento con menos empleados.

**INSERT INTO DEPART**

**SELECT 65, 'I+D', LOC**

**FROM DEPART**

**WHERE DEPT\_NO IN (SELECT DEPT\_NO**

**FROM EMPLE**

**GROUP BY DEPT\_NO**

**HAVING COUNT(\*) IN (SELECT MIN(COUNT(\*))**

**FROM EMPLE**

**GROUP BY DEPT\_NO));**

* 1. Insertar una avería en Averias\_Parque para la atracción Enterprise con fecha\_falla el día de ayer, fecha\_arreglo y coste a nulo y dni\_emple el encargado de la zona de la atracción.

**INSERT INTO AVERIAS\_PARQUE**

**SELECT AVERIAS\_PARQUE.COD\_ATRACCION, SYSDATE-1, NULL, NULL, DNI\_EMPLE**

**FROM AVERIAS\_PARQUE JOIN ATRACCIONES ON ATRACCIONES.COD\_ATRACCION = AVERIAS\_PARQUE.COD\_ATRACCION**

**WHERE NOM\_ATRACCION LIKE 'Enterprise';**

* 1. Insertar en ATRACCIONES una nueva at racción con el código ‘A140’, nombre ‘Que Miedo’, fecha de inauguración dentro de 30 días, capacidad 40 y nombre de zona la zona que menos atracciones tiene.

**INSERT INTO ATRACCIONES**

**SELECT 'A140', 'Que Miedo', SYSDATE+30, 40, NOM\_ZONA**

**FROM ATRACCIONES**

**GROUP BY NOM\_ZONA**

**HAVING COUNT(\*) IN (SELECT MIN(COUNT(\*))**

**FROM ATRACCIONES**

**GROUP BY NOM\_ZONA);**

1. **UPDATE**
   1. Modificar la dirección del alumno que se apellida Cerrato con ‘c/ Valbuena 87 1A’.

UPDATE ALUMNOS

SET DIREC = 'C/Valbuena 87 1A'

WHERE APENOM LIKE 'Cerrato%';

* 1. Modificar en la tabla averias\_parque la avería de la atracción ‘A100’ que tiene la fecha de arreglo a nulo poniendo dicha fecha al día de hoy y el coste de la avería a 1200.

UPDATE AVERIAS\_PARQUE

SET FECHA\_ARREGLO = SYSDATE,

COSTE\_AVERIA = 1200

WHERE COD\_ATRACCION LIKE 'A100';

* 1. Modificar la comisión de los empleados de departamentos de Barcelona poniéndoles 300€ a los que no tienen nada.

UPDATE EMPLE

SET COMISION = 300

WHERE DEPT\_NO IN (SELECT DEPT\_NO

FROM DEPART

WHERE LOC LIKE 'BARCELONA');

* 1. Cambiarle el oficio y el salario a TOVAR por el de ALONSO

UPDATE EMPLE

SET OFICIO = (SELECT OFICIO FROM EMPLE WHERE APELLIDO LIKE 'ALONSO'),

SALARIO = (SELECT SALARIO FROM EMPLE WHERE APELLIDO LIKE 'ALONSO')

WHERE APELLIDO LIKE 'TOVAR';

* 1. Subir 150€ a todos los empleados que empezaron a trabajar en la empresa antes del año 2012.

UPDATE EMPLE

SET SALARIO = SALARIO+150

WHERE TO\_CHAR(FECHA\_ALTA, 'YYYY') < 2012;

* 1. Subir 100€ a los empleados que ganan el mínimo salario en la empresa

UPDATE EMPLE

SET SALARIO = SALARIO+100

WHERE SALARIO IN (SELECT MIN(SALARIO) FROM EMPLE);

* 1. Subir 50€ a los empleados del departamento de INVESTIGACION que ganan menos de la media del departamento de INVESTIGACION.

UPDATE EMPLE

SET SALARIO = SALARIO+50

WHERE DEPT\_NO IN (SELECT DEPART.DEPT\_NO

FROM DEPART JOIN EMPLE ON EMPLE.DEPT\_NO = DEPART.DEPT\_NO

WHERE DNOMBRE LIKE 'INVESTIGACION')

AND SALARIO < (SELECT AVG(SALARIO)

FROM DEPART JOIN EMPLE ON EMPLE.DEPT\_NO = DEPART.DEPT\_NO

WHERE DNOMBRE LIKE 'INVESTIGACION');

1. **DELETE**

(CADA VEZ QUE REALICEIS UN BORRADO HACED ROLLBACK)

* 1. Borrar todos los empleados.

DELETE EMPLE;

* 1. Borrar los empleados de oficio Analista.

DELETE EMPLE

WHERE OFICIO LIKE 'ANALISTA';

* 1. Borrar los empleados del departamento de VENTAS

DELETE EMPLE

WHERE DEPT\_NO IN (SELECT DEPT\_NO

FROM DEPART

WHERE DNOMBRE LIKE 'VENTAS');

* 1. Borrar los empleados del departamento que menos empleados tiene.

DELETE EMPLE

WHERE DEPT\_NO IN (SELECT DEPT\_NO

FROM EMPLE

GROUP BY DEPT\_NO

HAVING COUNT(\*) IN (SELECT MIN(COUNT(\*)) FROM EMPLE GROUP BY DEPT\_NO));

* 1. Borrar los departamentos que no tienen empleados.

DELETE DEPART

WHERE DEPT\_NO NOT IN (SELECT DEPT\_NO FROM EMPLE);

* 1. Borrar al empleado más antiguo de la empresa

DELETE EMPLE

WHERE FECHA\_ALTA IN (SELECT(MIN(FECHA\_ALTA)) FROM EMPLE);