

1.

--Al crear las tablas correspondientes especificar la opción NOT NULL.

SQL

```
CREATE TABLE nombre_tabla (nombre_campo TIPO NOT NULL, ... ) ;
```

--Si la tabla ya estuviese creada:

```
ALTER TABLE nombre_tabla MODIFY nombre_campo TIPO NOT NULL
```

2.

```
ALTER TABLE empleados ADD CONSTRAINT ck_sexo CHECK (sexo ='H' OR sexo='M') ;
```

SQL

3.

```
ALTER TABLE departamentos ADD CONSTRAINT uk_nombre_dpto UNIQUE KEY (nombre_dpto) ;
```

SQL

```
ALTER TABLE trabajos ADD CONSTRAINT uk_nombre_trab UNIQUE (nombre_trab) ;
```

SQL

4.

```
ALTER TABLE historial_salarial ADD CONSTRAINT pk_historial_salarial PRIMARY KEY (empleado_dni, salario, fecha_comienzo) ;
ALTER TABLE historial_laboral ADD CONSTRAINT pk_historial_laboral PRIMARY KEY (empleado_dni, trabajo_cod, fecha_inicio) ;
```

SQL

5.

/\* CLAVES PRIMARIAS \*/

```
ALTER TABLE empleados ADD CONSTRAINT pk_empleados PRIMARY KEY (dni) ;
ALTER TABLE historial_salarial ADD CONSTRAINT pk_historial_salarial PRIMARY KEY (empleado_dni, salario, fecha_comienzo) ;
ALTER TABLE historial_laboral ADD CONSTRAINT pk_historial_laboral PRIMARY KEY (empleado_dni, trabajo_cod, fecha_inicio) ;
ALTER TABLE departamentos ADD CONSTRAINT pk_departamentos PRIMARY KEY (departamento_cod) ;
ALTER TABLE estudios ADD CONSTRAINT pk_estudios PRIMARY KEY (empleado_dni, universidad, especialidad) ;
ALTER TABLE universidades ADD CONSTRAINT pk_universidades PRIMARY KEY (uni_cod) ;
ALTER TABLE trabajos ADD CONSTRAINT pk_trabajos PRIMARY KEY (trabajo_cod) ;
```

SQL

/\* CLAVES AJENAS \*/

```
ALTER TABLE historial_salarial ADD CONSTRAINT fk_historial_salarial_empleado FOREIGN KEY (empleado_dni) REFERENCES empleados (dni) ;
ALTER TABLE historial_laboral ADD CONSTRAINT fk_historial_laboral_empleado FOREIGN KEY (empleado_dni) REFERENCES empleados (dni) ;
ALTER TABLE historial_laboral ADD CONSTRAINT fk_historial_laboral_supervisor FOREIGN KEY (supervisor_dni) REFERENCES empleados (dni) ;
ALTER TABLE historial_laboral ADD CONSTRAINT fk_historial_laboral_trabajo FOREIGN KEY (trabajo_cod) REFERENCES trabajos (trabajo_cod) ;
ALTER TABLE historial_laboral ADD CONSTRAINT fk_historial_laboral_dpto FOREIGN KEY (dpto_cod) REFERENCES departamentos (dpto_cod) ;
ALTER TABLE departamentos ADD CONSTRAINT fk_departamento_padre FOREIGN KEY (dpto_padre) REFERENCES departamentos (dpto_cod) ;
ALTER TABLE estudios ADD CONSTRAINT fk_estudios_empleado FOREIGN KEY (empleado_dni) REFERENCES empleados (dni) ;
ALTER TABLE estudios ADD CONSTRAINT fk_estudios_universidad FOREIGN KEY (universidad) REFERENCES universidades (univ_cod) ;
```

## Realizar todas las operaciones

1.

Insertar dos filas cumpliendo las restricciones

2.

```sql

```
INSERT INTO empleados (nombre, apellidol, apellido2, dni, sexo )
VALUES ('Sergio', 'Palma', 'Entrena', '111222', 'P') ;
```

```
INSERT INTO empleados (nombre, apellidol, apellido2, dni, sexo)
VALUES ('Lucia', 'Ortega', 'Plus', '222333', NULL) ;
```

Puesto que existe una restricción para los valores del campo Sexo (H y M) ninguna de las sentencias anteriores funcionaría.

```
INSERT INTO historial_laboral (empleado_dni, trab_cod, fecha_inicio, fecha_fin, dpto_cod, supervisor_dni )VALUES ('111222', NULL, '6JUN
1996', NULL, NULL, '222333') ;
```

SQL

3.

Se produciría un error por violación de la clave ajena definida sobre la tabla historial\_laboral que afecta al campo supervisor\_dni que referencia al campo dni de la tabla empleados.

4.

No se podrán borrar aquellas Universidades cuyo código esté presente en la tabla Estudios por existir una restricción de clave ajena desde dicha tabla hacia la tabla Universidades. Es decir, no pueden borrarse registros padres mientras existan registros hijos dependientes. Una manera de poder borrar sería eliminando la clave ajena:

```
ALTER TABLE estudios DROP CONSTRAINT fk_estudios_universidades;
```

SQL

También se podría modificar la restricción para que ponga un null en los registros hijos en el caso de que se borre el registro padre:

```
ALTER TABLE estudios ADD CONSTRAINT fk_estudios_universidad FOREIGN KEY (universidad) REFERENCES universidades (univ_cod) ON DELETE SET NULL;
```

SQL

También se podría haber creado la FK con la opción “ON DELETE CASCADE”

1. No se hace

6.

```
ALTER TABLE empleados ADD valoracion NUMBER DEFAULT 5 ;  
ALTER TABLE empleados ADD CONSTRAINT ck_valoracion CHECK (valoracion BETWEEN 1 AND 10) ;
```

SQL

7.

```
ALTER TABLE empleado MODIFY nombre VARCHAR(10) NULL ;
```

SQL

8.

```
ALTER TABLE empleados MODIFY direccion VARCHAR(40) ;
```

SQL

9.

Se pueden realizar cambios en los tipos de datos siempre que el contenido de los campos que se pretenden modificar cumplan o sean compatibles con los nuevos tipos de datos.

10.

```
ALTER TABLE empleados DROP CONSTRAINT pk_empleados ;  
ALTER TABLE empleados ADD CONSTRAINT pk_empleados PRIMARY KEY (nombre, apellido1, apellido2) ;
```

SQL

Si alguna tabla tuviera una clave ajena apuntando a la clave primaria de empleados, no sería posible crear la nueva clave primaria sin antes haber eliminado dichas claves ajenas.