

Oracle 11g SQL

Curso práctico de formación

Antolín Muñoz Chaparro



Oracle 11g SQL. Curso práctico de formación
Antolín Muñoz Chaparro

ISBN: 978-84-938312-4-0

EAN: 9788493831240

Copyright © 2011 RC Libros
© RC Libros es un sello y marca comercial registrados

Oracle 11g SQL. Curso práctico de formación.

Reservados todos los derechos.

Ninguna parte de este libro incluida la cubierta puede ser reproducida, su contenido está protegido por la Ley vigente que establece penas de prisión y/o multas a quienes intencionadamente reprodujeren o plagiaren, en todo o en parte, una obra literaria, artística o científica, o su transformación, interpretación o ejecución en cualquier tipo de soporte existente o de próxima invención, sin autorización previa y por escrito de los titulares de los derechos de la propiedad intelectual.

RC Libros, el Autor, y cualquier persona o empresa participante en la redacción, edición o producción de este libro, en ningún caso serán responsables de los resultados del uso de su contenido, ni de cualquier violación de patentes o derechos de terceras partes. El objetivo de la obra es proporcionar al lector conocimientos precisos y acreditados sobre el tema tratado pero su venta no supone ninguna forma de asistencia legal, administrativa ni de ningún otro tipo, si se precisase ayuda adicional o experta deberán buscarse los servicios de profesionales competentes. Productos y marcas citados en su contenido estén o no registrados, pertenecen a sus respectivos propietarios.

RC Libros
Calle Mar Mediterráneo, 2. Nave 6
28830 SAN FERNANDO DE HENARES, Madrid
Teléfono: +34 91 677 57 22
Fax: +34 91 677 57 22
Correo electrónico: info@rclibros.es
Internet: www.rclibros.es

Diseño de colección, cubierta y pre-impresión: Grupo RC
Impresión y encuadernación: Service Point
Depósito Legal: M-
Impreso en España

15 14 13 12 11 (1 2 3 4 5 6 7 8 9 10 11 12)

PRÓLOGO

Corría el año 1990 cuando entré en la Administración del Estado del Gobierno español. Tenía 19 años y acababa de terminar los estudios secundarios. Era mi primer trabajo.

Un año antes ni siquiera podía imaginar que acabaría trabajando en la Administración, y tan joven. Estaba terminando el ciclo de estudios secundarios, en concreto COU y toda mi ilusión era poder concluir con buena nota los estudios para que la media con la Selectividad, me pudiera dar una nota suficiente para cursar una carrera en la facultad de Biológicas. Siempre me había atraído esta rama de la ciencia y dentro de ella me apasionaba la genética. Estaba convencido de querer convertirme en un futuro científico que investigase, en un laboratorio, todo lo relacionado con esta materia.

Sin embargo, las cosas se truncaron al finalizar el curso. Una asignatura lo cambió todo. ¡Maldita física pensé por entonces!, aunque tampoco se libró de ciertos apelativos poco apropiados la profesora que decidió que tenía que repetir esa asignatura de COU y por lo tanto no podía presentarme a Selectividad. Esto me produjo una decepción profunda, que quizás a otra persona, le hubiera llevado cerca de la depresión, pero me considero una persona activa y práctica, así que decidí ocupar el mucho tiempo sobrante que tenía durante el día para encauzar mi vida. Me apunté a una academia y me empecé a preparar unas oposiciones a la Administración, a la vez realicé un curso sobre programas de aplicación a la gestión

administrativa donde tuve un fugaz contacto con la informática pero que fue suficiente para encender en mi interior la llama del interés por esta rama. Y por supuesto por las noches a estudiar física en el instituto.

Todo este cóctel de actividades tuvo sus frutos en el verano de 1989, fecha clave en mi vida porque supuso aprobar unas oposiciones, aprobar la famosa asignatura de física y también hacer lo propio con la Selectividad. Con un año de retraso llegaba la posibilidad de iniciar unos estudios universitarios, pero la gran puerta que se me abrió con la entrada a trabajar en la Administración hizo que cambiase de rumbo. La semilla que se había plantado en mí respecto a la informática me llevó a compatibilizar el trabajo con los estudios de Formación Profesional de II grado en Informática. Aparentemente era un paso atrás en cuanto a formación porque son estudios equiparables a los del bachillerato, pero para mí supuso un aprendizaje esencial en la rama de la Informática que me trasladó definitivamente a este mundo. Olvidé la idea inicial de estudiar en la facultad de Biológicas. Más tarde también acabé obteniendo la diplomatura en Ingeniería Técnica en Informática de Gestión.

Pero sin duda lo que soy actualmente y sobre todo, los conocimientos que he adquirido durante todos estos años en los productos de Oracle, se los debo a la Administración y en particular a una persona que me acogió como un hijo en el departamento de Informática de lo que era por entonces la Dirección General de Incentivos Regionales. Aunque ha sido mi jefe durante 20 años, por encima de todo ha sido un compañero más, un amigo y casi un padre. Pese a mi inexperiencia en el sector de las Tecnologías de la Información, desde el primer momento me dio su confianza y sobre todo me infundió la tranquilidad necesaria para poder llevar a cabo un aprendizaje paulatino practicando en el mundo real, es decir, pudiendo aportar mi granito de arena en forma de programación para las aplicaciones en producción que por entonces había. Eran tiempos en los que los ordenadores centrales o mainframes eran grandes armarios llenos de bobinas de cintas magnéticas, aún se utilizaban las tarjetas perforadas, no existían los ordenadores personales o al menos no habían llegado a la Administración, y se trabajaba casi tanto con las máquinas de escribir como con los famosos "terminales tontos"; aquellos monitores con emuladores para acceso a los sistemas operativos de los mainframes (en general emuladores VT), que únicamente tenían un teclado presentando la información en modo carácter sobre pantallas negras con textos verdes muy luminosos. Eran los tiempos de los sistemas operativos propietarios; es decir, cada empresa fabricante de un mainframe incorporaba su propio sistema operativo que no funcionaba en ningún otro equipo. También eran los tiempos de la programación en COBOL.

Con los años me he dado cuenta de que tuve la fortuna de poder trabajar con una persona que fue un adelantado a su época en usar las nuevas tecnologías que aparecían en el mercado cuando quizás lo más cómodo y sensato para no buscarse problemas era el continuismo, también conocido como "no meneallo"; si algo funciona para qué cambiarlo. Pero por esta regla de tres hoy en día todavía seguiríamos programando en COBOL y utilizando terminales "tontos" en la Administración. Este no era el carácter de mi jefe, ¡al contrario!, siempre fue valiente en sus apuestas informáticas, y no le tembló el pulso para realizar cambios que se adaptasen a las nuevas tecnologías y tendencias del sector. Así en el año 1991 jubiló el mainframe y adquirió un nuevo servidor con sistema operativo estándar Unix System V, con una base de datos Oracle versión 6. Aunque Oracle llevaba funcionando en el mercado desde el año 1983 con la versión 3 de base de datos, y empezaba a despuntar entre las empresas fabricantes de sistemas gestores de base de datos relacionales, aún no era líder en el sector y mucho menos había irrumpido en la Administración. Así que sin ningún género de dudas puedo decir que mi jefe fue un pionero en el uso de Oracle dentro de la Administración y fruto de ello yo pude especializarme en sus productos, cuando por entonces muy pocas empresas externas tenían gente formada que pudiese programar contra este tipo de base de datos. Pero lo más importante es que pese a que ya por entonces en la Administración se contrataba a gente externa para la programación de las aplicaciones, mi jefe apostó por su gente, por el personal propio de la Administración que trabajaba en su departamento, entre los que me encontraba yo, para desarrollar una nueva aplicación en SQL*FORMS V.2.3 contra la base de datos Oracle versión 6, que sustituyese a la que había por entonces desarrollada en COBOL. Apostó con valentía, dándonos el tiempo necesario para formarnos en los productos de Oracle y luego la tranquilidad para desarrollar la aplicación codo con codo con los usuarios. En el fondo el que mejor puede conocer las necesidades de un usuario es la persona que está día a día tratando con los problemas cotidianos que le van surgiendo, conoce sus inquietudes y sugerencias, y al final tiene un conocimiento más amplio y profundo del funcionamiento del organismo.

Así que en poco más de un año tuvimos implementada la nueva aplicación, cuando la previsión más optimista de las empresas externas, para hacer este mismo trabajo, era de tres.

A partir de ahí siguió la evolución tecnológica en hardware y software, en paridad con la aparición de nuevos productos y/o avances tecnológicos. En sistemas operativos de Microsoft conocí, trabajé y adquirí práctica en: Windows 3.0, 3.1, 95, 98, NT 3.5, NT 4.0, 2000, XP, 2003 y Vista, hasta llegar al actual Windows 7. En cuanto al Unix: Unix System V y Solaris 8. Y por último en lo que respecta a Oracle, adquirí los conocimientos y práctica necesaria para llegar a escribir este libro, administrando

y desarrollando contra las siguientes versiones de base de datos 6, 7, 8, 8i, 9i y 10g, hasta llegar a la versión actual 11g. Igualmente en el desarrollo de aplicaciones he trabajado con las siguientes herramientas y productos de Oracle: Forms 2.3, Oracle Developer 1, Oracle Developer 6.0 y 6i, Oracle 9iDS y Oracle Developer Suite 10g.

Todo el conocimiento adquirido durante estos años me ha permitido formar a otras personas en el uso de los productos de Oracle, habiendo impartido diversos cursos tanto de administración como de desarrollo. Este libro refunde gran parte del material utilizado durante los cursos que he desarrollado en diversas academias de formación y empresas privadas, sirviendo de referencia y ayuda a quien quiera iniciar un aprendizaje en el lenguaje SQL, o mejorar el nivel de conocimientos sobre el mismo. La información que encontrará en este curso se encuentra actualizada hasta la versión 11g Rel. 2., de la base de datos de Oracle.

Pensando en un amplio sector de la población que no puede acudir de forma presencial a la distinta oferta formativa que se ofrece para este lenguaje, este libro le aporta cómodamente un curso planificado equiparable a los impartidos de forma presencial, con un pautado ritmo de aprendizaje y que siempre servirá como guía de referencia para posteriores consultas.

Han pasado más de dos décadas desde que maldije aquella fatalidad que supuso para mí repetir la asignatura de física de COU, y hoy en día solo puedo decir que doy gracias al destino, y a la profesora que dio un giro a mi vida porque si no hubiese ocurrido aquello, hoy no estaría aquí escribiendo este libro. No quiero olvidarme tampoco de agradecer los 20 años que pasé con la que para mí es la "Gran Familia de Incentivos", que permitió formarme como profesional de la Informática y que me ayudó a modelar mi aspecto humano; para todos ellos va dedicado este libro, con un recuerdo especial para mis últimos compañeros y amigos en el departamento de Informática: Lourdes, Juan, Araceli, Chema y por supuesto para el que ha sido mi mentor y a quien va dedicado especialmente este prólogo: Eduardo S.

Los archivos para la resolución de los supuestos se encuentran disponibles en la página individual del libro, accediendo a la web: www.rclibros.es

CONCEPTOS DE BASES DE DATOS RELACIONALES

1

¿QUÉ ES UNA BASE DE DATOS?

Es un conjunto de información almacenada en una estructura de ficheros en disco, que será consultada y modificada por los usuarios que acceden a la misma, siendo el Sistema Gestor de Base de Datos (en adelante SGBD) el encargado de llevar físicamente a cabo estas operaciones.



Gráficamente, una base de datos se representa con un cilindro.

Dentro del mismo se ubicará la información que contenga.

El proceso de consulta de una base de datos quedaría representado con la *figura 1-1*, donde un usuario ejecuta desde su ordenador una sentencia (`select * from tabla`), contra la base de datos.



Fig. 1-1 Ejecución de una sentencia de consulta.

La base de datos convierte la lógica de la sentencia ejecutada por el usuario en una búsqueda física de la información dentro de la base de datos. Para ello, tendrá que acceder a los índices que contenga la tabla consultada, para ubicar la zona física del disco donde se encuentra la información.

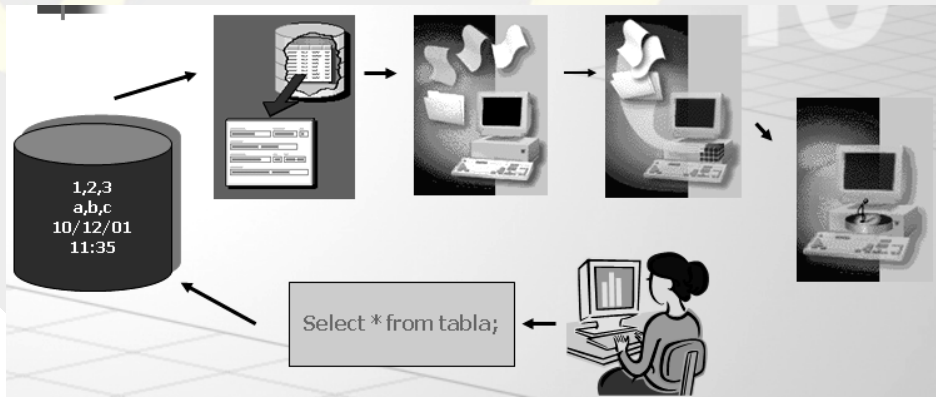


Fig. 1-2 Búsqueda física de la información en la base de datos.

Completada la búsqueda de la información física, el SGBD devuelve la información al usuario de forma estructurada, según el tipo de consulta que haya realizado.

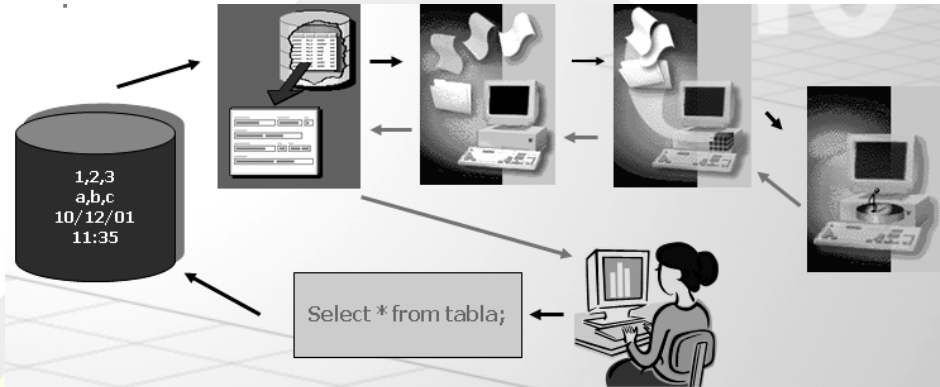


Fig. 1-3 Búsqueda física de la información en la base de datos.

¿QUÉ SIGNIFICA BASE DE DATOS RELACIONAL?

Cuando hablamos de base de datos, tendemos a unir el concepto relacional a su significado, pero realmente; ¿qué significa base de datos relacional? Y ¿cuándo podemos considerar a una base de datos como relacional?

- Una base de datos relacional es aquella que se fundamenta o utiliza las teorías del modelo relacional de Codd.
- Oracle implementa el modelo relacional en sus SGBD desde su primera versión. Actualmente ha extendido este modelo (sin abandonarlo), para añadir funcionalidades de los entornos orientados a objeto.
- Otros SGBD que utilizan el modelo relacional son SQLSERVER, SYBASE y MySQL.

INTRODUCCIÓN AL MODELO RELACIONAL

El modelo relacional fue enunciado por Codd a finales de los años 60.

Está basado en la teoría de las relaciones, donde se contemplan como elementos principales: las entidades y las relaciones.

El elemento básico es la **relación**: nexo entre 2 o más entidades.

Elementos del modelo relacional

Los elementos del modelo relacional son:

- Entidades.
- Relación.
- Atributos.
- Tuplas.
- Dominios.

ENTIDADES

Una entidad es un objeto del mundo real que tiene interés para el sistema, y del cual se puede extraer una serie de atributos.

Ejemplos de entidades son: una casa, un coche, una oficina, un hospital, una persona, etc.

RELACIONES

Una relación es el nexo (vínculo de unión) entre 2 o más entidades a través de alguno/s de sus atributos.

Por ejemplo, si tomamos como referencia 2 entidades: cliente y factura, ambas se pueden relacionar por el código de cliente (atributo perteneciente a ambas entidades).

ATRIBUTOS

Un atributo es la característica o propiedad que define una entidad, y que la diferencia de otra.

Ejemplos de atributos para la entidad casa serían: ubicación, número de habitaciones, metros cuadrados, tipo de vivienda, etc.

TUPLAS

Una tupla es la unión de todos los atributos de una entidad en una sola estructura independiente.

Un ejemplo de tupla sería: Pepito, Pérez, 26, C/ Madrid, 913332211

DOMINIOS

Un dominio es un conjunto de valores que puede adoptar un atributo de una entidad.

Un ejemplo del dominio correspondiente al atributo metros cuadrados de una casa sería: $1 \text{ m}^2 - 30 \text{ m}^2$, $31 \text{ m}^2 - 60 \text{ m}^2$, $61 \text{ m}^2 - 100 \text{ m}^2$, $100 \text{ m}^2 - 200 \text{ m}^2$, $> 200 \text{ m}^2$.

Tipos de clave en el modelo relacional

Una clave es un conjunto de valores de una entidad que distingue unívocamente esta información de otro conjunto de valores de la misma entidad.

Por ejemplo, si consideramos una entidad alumno que incluya como atributos: DNI y nombre de alumno, la clave de la misma sería el DNI, que es el único dato que hace diferenciar un alumno de otro.

En el modelo relacional se consideran 3 tipos de claves:

- Primaria.
- Alternativa.
- Ajena.

CLAVE PRIMARIA

Una clave primaria corresponde con el/los atributos de una entidad cuyo conjunto de valores distingue unívocamente una tupla de otra.

En el modelo relacional solo se permite una clave primaria por cada entidad.

Un ejemplo de clave primaria para la entidad trabajador sería el DNI.

CLAVE ALTERNATIVA

Una clave alternativa es aquel atributo o conjunto de atributos que podrían haber formado una clave primaria, pero que al ya existir una definida, no pueden hacerlo.

Un ejemplo de clave alternativa para la entidad trabajador sería el número de la seguridad social.

CLAVE AJENA

Una clave ajena es aquel atributo o conjunto de atributos de una entidad que se relaciona con la clave primaria de otra entidad.

Por ejemplo, si tomamos como referencia 2 entidades: cliente y factura, que tienen como atributo en común el DNI de un cliente, podemos definir como clave primaria de cliente el atributo DNI y como clave ajena de factura también el mismo atributo de DNI. De esta forma podemos relacionar factura con cliente.

Restricciones en el modelo relacional

En el modelo relacional se permite la determinación de restricciones sobre los atributos de una entidad.

Una restricción se define sobre un atributo y consiste en una limitación que se le aplica al mismo.

Una clave es por definición una restricción que limita la repetición de valores para la clave (en el caso de las primarias o alternativas), o la asignación de valores distintos a los de la clave primaria relacionada (para el caso de las claves ajenas).

Así mismo, un dominio definido sobre un atributo también lo es, porque limita los valores posibles que se pueden almacenar.

Por último, también se pueden definir restricciones que no sean del tipo clave o del tipo dominio. Por ejemplo, cuando queremos definir que un atributo concreto de una entidad no pueda almacenar valores vacíos.

Teorías de normalización

Son normas y reglas que permiten convertir un modelo conceptual (Entidad/Relación) en un modelo relacional.

La aplicación de estas teorías nos permite convertir el modelo relacional en un modelo físico de construcción de la base de datos, donde existirán elementos propios del tipo: tablas, columnas, índices y constraints.

RELACIÓN ENTRE LOS ELEMENTOS DEL MODELO RELACIONAL Y UNA BASE DE DATOS

Para poder utilizar el modelo relacional (diagrama de Entidad/Relación) en un SGBD, es necesaria la conversión de los elementos del modelo relacional a elementos propios de una base de datos relacional.

A continuación, se muestra una tabla con la conversión de elementos:

Modelo conceptual Entidad/Relación (Modelo relacional)	Base de datos
Entidades	Tablas
Atributos	Columnas
Tuplas	Filas
Dominios	Restricciones
Claves	Restricciones e índices

Una **tabla** es un conjunto de filas y columnas donde se almacena la información.

Una **columna** representa un único valor o atributo de una tabla.

Una **fila** representa la unión de un conjunto de valores repartidos entre las columnas de la tabla.

Una **restricción (constraint)** en inglés, es aquella limitación que se impone a los valores que puede llegar a tomar una columna de una tabla. Así mismo se pueden imponer restricciones a nivel de toda la tabla, usando claves.

Un **índice** es una estructura de acceso rápido que utiliza el SGBD para localizar más rápido la información física contenida en la base de datos.

COMPOSICIÓN DE UN ÍNDICE

Un índice se crea para poder acceder a la información física de manera más rápida por parte del SGBD, y se compone de un conjunto de columnas que son la clave de acceso a través del índice + la identificación de la posición física dentro del fichero, donde se encuentra la fila para recuperar la información correspondiente.

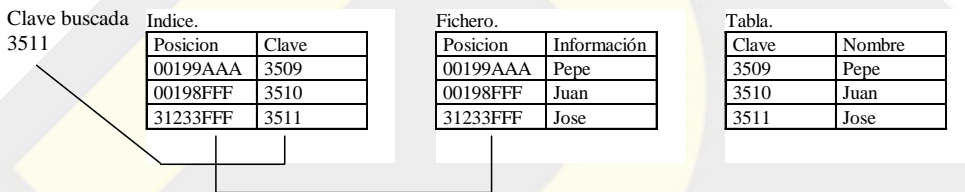


Fig. 1-4 Esquema de un índice y método de localización del dato físico.

CREACIÓN DE UN DIAGRAMA RELACIONAL

En este apartado se orienta sobre la creación de un diagrama relacional, los pasos a seguir y los elementos a contemplar en el mismo.

La teoría del modelo de Codd parte de la idea de creación de un diagrama Entidad/Relación, donde únicamente se especificarán las Entidades (con sus atributos clave y no clave), y las Relaciones entre las mismas, especificando cardinalidades y grado de las relaciones. Este tipo de diagramas queda fuera del estudio de este curso; no obstante, al final de este apartado se muestra el diagrama Entidad/Relación del que se partiría para poder llegar a obtener el modelo relacional descrito a continuación como práctica de este curso.

El supuesto práctico que se expresa en las distintas fases de este apartado quiere reflejar, en un modelo relacional, la estructura de un comercio que vende productos informáticos. Evidentemente esta descripción es muy abierta, por lo que el diagrama que se propone podría ser diferente dependiendo de la interpretación que quiera darle cada alumno a este enunciado.

Fase 1. Definición de entidades

En la primera fase del diseño del diagrama tenemos que identificar las entidades, sus atributos y las restricciones que pueden imponerse sobre las mismas, si las hubiera.



Fig. 1-5 Esquema de entidades de un comercio que vende productos informáticos.

Fase 2. Identificación de claves primarias y alternativas

En la segunda fase hay que definir los atributos de cada entidad que forman la clave primaria y si las hubiera, las posibles claves alternativas.



Fig. 1-6 Claves primarias de las entidades.

Fase 3. Identificación de claves ajenas y relaciones

En la última fase del diseño hay que definir las claves ajenas que relacionan unas entidades con otras y los atributos que se ven afectados por las mismas.

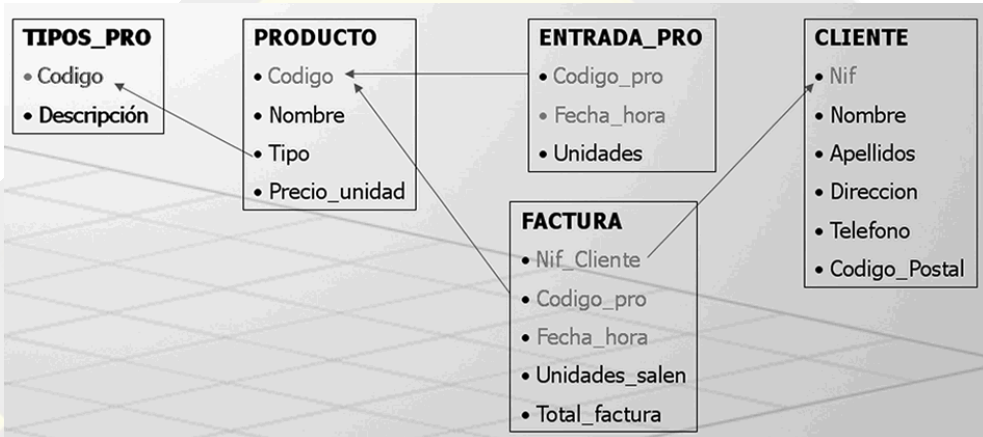


Fig. 1-7 Diagrama relacional completo.

Diseño Entidad/Relación

Como se ha comentado al comienzo del apartado, un diseño relacional requiere de un diseño Entidad/Relación previo. Para nuestro supuesto, el diagrama Entidad/Relación sería el siguiente:

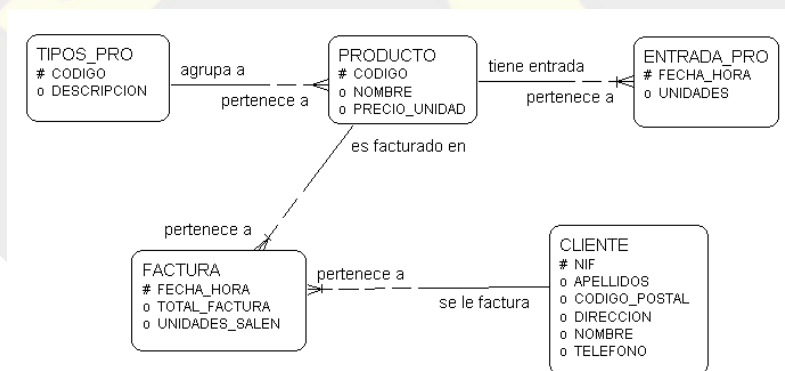
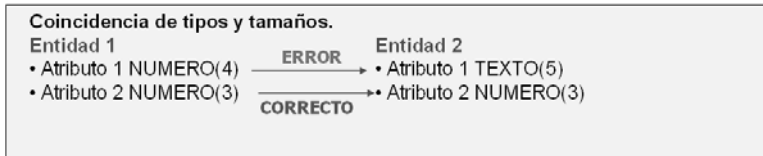


Fig. 1-8 Modelo Entidad/Relación de un comercio que vende productos informáticos.

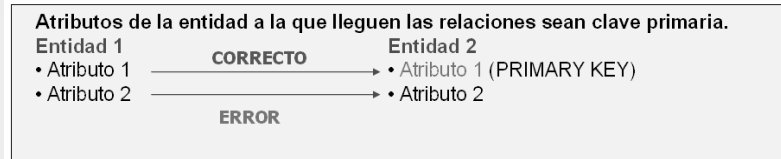
REGLAS PARA LA CREACIÓN DE CLAVES AJENAS

Para crear una relación entre entidades o claves ajenas, es necesario que se cumplan una serie de condiciones:

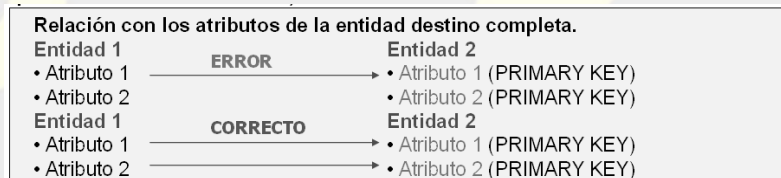
- Que el tipo de datos y tamaño entre los atributos coincida.



- Que el/los atributos de la entidad destino de la relación sean clave primaria o clave única.



- La relación con la entidad destino debe ser completa (con todos los atributos que forman la clave primaria de la misma).



OPERACIONES DE CONSULTA

En el modelo relacional de Codd se definen las siguientes operaciones de consulta sobre entidades:

- Selección.
- Proyección.
- Unión.
- Intersección.
- Diferencia.
- Producto cartesiano.
- Join.

Operación de selección

Es una operación de consulta que obtiene un conjunto de tuplas de una sola entidad con o sin condiciones.

ATRI1	ATRI2	ATRI3	ATRI4	ATRI5
1	2	3	4	6
7	8	9	10	11
12	13	14	15	16

Fig. 1-9 Resultado de una operación de selección.

Operación de proyección

Es una operación de consulta que permite obtener valores de uno o varios atributos de una sola entidad.

ATRI1	ATRI2	ATRI3	ATRI4	ATRI5
1	2	3	4	6
7	8	9	10	11
12	13	14	15	16

Fig. 1-10 Resultado de una operación de proyección.

Operación de unión

Para poder realizar una operación de consulta de unión entre entidades, tiene que coincidir el número de atributos de ambas y el tipo.

El resultado de esta operación es la suma de tuplas de ambas entidades para los atributos que se unen.

A	B	C		A	D	C	=	A	C
1	1	11	∪	4	467	44		1	11
2	2	22		5	578	55		2	22
3	3	33		6	678	66		3	33
								4	44
								5	55
								6	66

Fig. 1-11 Resultado de una operación de unión.

Operación de intersección

Para realizar una operación de consulta con intersección entre entidades, tiene que coincidir el número de atributos de ambas y el tipo.

El resultado de esta operación es la aparición únicamente de las tuplas con el mismo contenido de ambas entidades.

A	B	C	∩	A	B	C	=	A	B	C
1	22	333		1	22	333		1	22	33
4	55	666		7	88	999				

Fig. 1-12 Resultado de una operación de intersección.

Operación de diferencia

Para realizar una operación de consulta con diferencia entre entidades, tiene que coincidir el número de atributos de ambas y el tipo.

El resultado de esta operación es la aparición de aquellas tuplas de la primera entidad que no coincidan con las tuplas de la segunda.

A	B	C	-	A	B	C	=	A	B	C
1	22	333		1	22	333		4	55	666
4	55	666		7	88	999				

Fig. 1-13 Resultado de una operación de diferencia.

Operación de producto cartesiano

La operación de consulta del producto cartesiano permite obtener un resultado que será una tupla por cada combinación entre cada tupla de la primera entidad y todas las tuplas de la segunda.

A	B	C	×	D	E	=	A	B	C	D	E
1	2	3		44	55		1	2	3	44	55
				66	77		1	2	3	66	77

Fig. 1-14 Resultado del producto cartesiano.

Operación de Join

La operación de consulta Join permite unir tuplas de dos entidades a través de algún atributo en común.

El resultado de esta operación es la suma de los atributos que se quieren mostrar.

A	B	C	J	A	E	F	=	A	B	C	E	F
1	22	33		1	565	32		1	22	33	565	32
				2	751	12						

Fig. 1-15 Resultado de una operación de Join.

SUPUESTO PRÁCTICO 0: Resolución en el Anexo I.

Diseñar un esquema de modelo relacional en el que se vean reflejados las entidades, atributos mínimos, relaciones y las claves primarias, alternativas y ajenas que considere oportunas, de acuerdo con el enunciado que se especifica a continuación. Opcionalmente, podrá diseñar, previo al modelo relacional, el modelo Entidad/Relación del que se derivaría el modelo relacional.

- Representar la estructura de un centro de enseñanza que posee delegaciones en varias provincias de España.
- Será necesario reflejar para el centro de enseñanza los siguientes atributos:

- Código de 1 a 99.
- Nombre.
- Dirección.
- Provincia.
- Teléfono.
- Código postal.
- Para las provincias habrá que reflejar:
 - Código provincia de 1 a 52.
 - Nombre.
- El código del centro es unívoco y distinto para todos los centros estatales.
- En cada centro se imparten una serie de cursos con los siguientes atributos:
 - Código de curso de 1 a 99999.
 - Nombre de curso.
 - Código del centro donde se imparte.
- Cada curso es impartido por una serie de profesores cualificados con los siguientes atributos:
 - DNI.
 - Nombre.
 - Apellidos.
 - Dirección.
 - Teléfono.
 - Código postal.
 - Código de curso.
 - Código de centro.
 - Nº de la Seguridad Social.
- El código del curso puede repetirse en centros distintos, por lo que no es unívoco. Un curso de un centro solo puede ser impartido por un profesor.
- Por último, a estos cursos asisten una serie de alumnos con los siguientes atributos:
 - DNI.
 - Nombre.
 - Apellidos.
 - Dirección.
 - Teléfono.
 - Código postal.
 - Código de curso.

- Código de centro.
- Fecha de comienzo del curso.
- Fecha de fin del curso.
- Un alumno en un mismo centro puede asistir a varios cursos, pero al mismo curso solo podrá asistir en fechas de comienzo distintas.