

BASE DE DATOS I

El modelo de Base de Datos Relacional





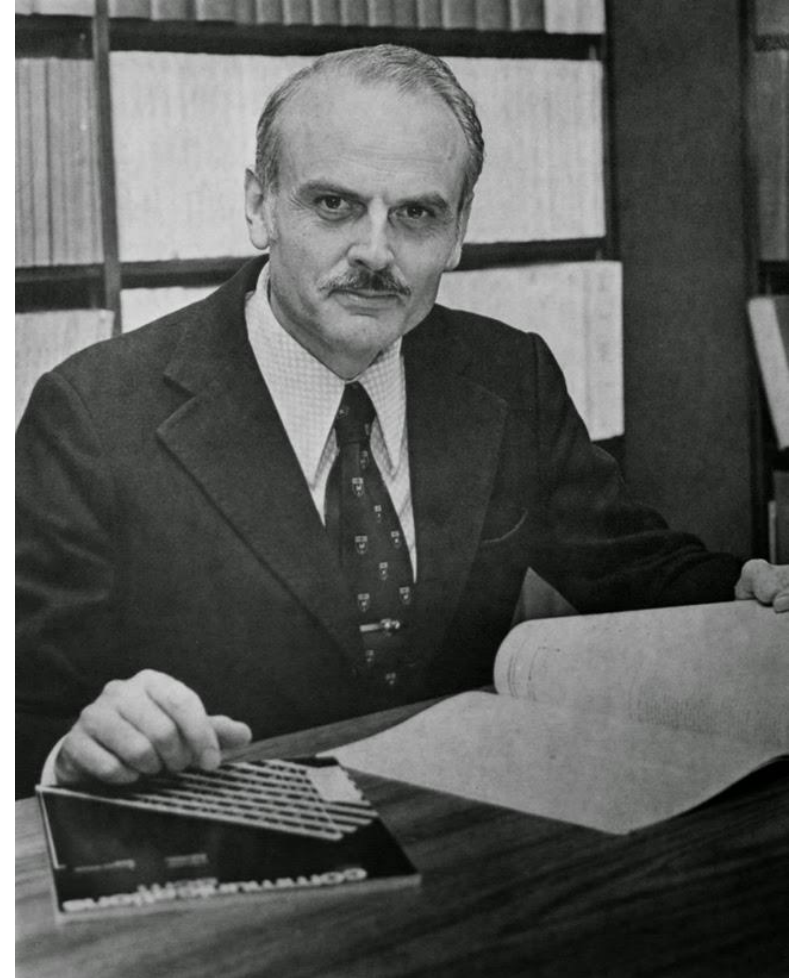
Introducción



Introducción



- El modelo relacional fue propuesto por E.F. Codd (1970) en su artículo original titulado “A relational model of data for large share data banks”. Está fundamentado en el concepto de una relación matemática y su base teórica, en la teoría de conjunto y la lógica de predicados de primer orden.
- En el modelo relacional los datos se encuentran almacenados en tablas (registros).



Recordemos



- Crear agrupaciones en supertipos y subtipos ¿qué ventajas ofrece?
- ¿Es posible crear la Entidad “Empleado” y “Administrativo” como subtipos de “Personal” si es que ambas entidades comparten los mismos atributos?
- La entidad de quien se hereda atributos y relaciones se llama...
- Dada una relación de jerarquía (herencia) como se puede implementar en el modelo relacional.



Podría Ud. describir...

... Por qué en una base de datos son importantes la integridad de entidad y la referencial





Al término de la sesión, el estudiante elabora un modelo relacional, a partir de un Diagrama Entidad Relación, mostrando mínimamente las relaciones, atributos y tuplas, verificando que se cumpla las reglas de integridad.

Desarrollo del tema



Una vista lógica de los datos



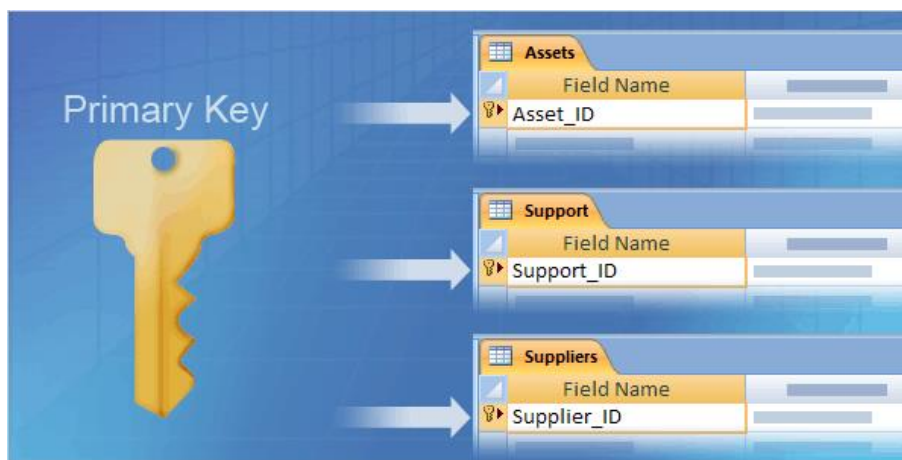
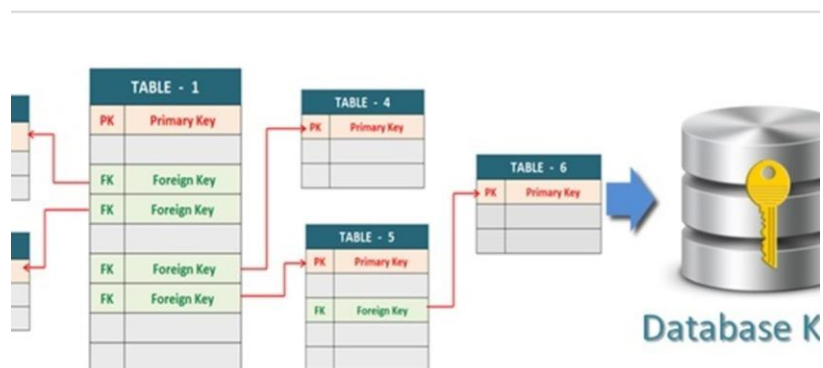
- El modelo de base de datos relacional permite una representación lógica de los datos y sus relaciones
- La simplicidad lógica produce metodologías de diseño de bases de datos simples y efectivas
- Ayudado por la creación de relaciones de datos basadas en una construcción lógica llamada relación



Características de una tabla relacional



- Una tabla es percibida como una estructura bidimensional compuesta de renglones y columnas.
- Cada renglón (o fila) de tabla (tupla) representa una ocurrencia única de entidad dentro del conjunto de entidades.
- Cada columna de tabla representa un atributo y cada columna tiene un nombre distinto.
- Cada intersección de renglón/columna representa un valor único de datos.
- Todos los valores de una columna deben apegarse al mismo formato de datos.
- Cada columna tiene un intervalo específico de valores conocido como dominio de atributo.
- El orden de los renglones y las columnas no tiene importancia para el DBMS.
- Cada tabla debe tener un atributo o una combinación de atributos que identifique de manera única a cada renglón



- Consiste en uno o más atributos que determinan otros atributos
- Son usados para :
 - Asegúrese de que cada fila de una tabla sea identificable de forma única
 - Establecer relaciones entre tablas y garantizar la integridad de los datos
- Llave principal (PK): atributo o combinación de atributos que identifica de forma única cualquier fila dada

Llaves: Determinación



- Estado en el que conocer el valor de un atributo permite determinar el valor de otro
- Es la base para establecer el papel de una clave
- Basado en las relaciones entre los atributos
- La notación breve para “A determina B” es $A \rightarrow B$.

EMPLEADO ← Nombre de la Relación

Clave Primaria	pasaporte	pnombre	apaterno	apmaterno	fono	fnacimiento	Atributos
	12095444	Alberto	Gómez	Martínez	2345676	20/11/1969	Tuplas
	9509590	Luisa	Jordán	Soto	3344567	12/09/2000	
	19456873	Cristian	Muñoz	Pereira	4567912	12/10/2010	
	20345765	Josefina	Carvajal	Durán	3456835	05/06/2011	
	15687490	Marcos	Ramírez	Ponce		28/02/1978	

ID Estudiante	Semestre	Clase	Profesor	Monitor
1234	6	Numerical Methods	Ixent	John
1221	4	Numerical Methods	Ixent	Robert
1234	6	Visual Computing	Olmer	Bob
1201	2	Numerical Methods	Sebastian	Peter
1201	2	Physics II	Edgar	Simon

ID Estudiante → Semestre
Clase → Profesor
ID Estudiante, Clase → Monitor

Llaves: dependencias



dependencia funcional

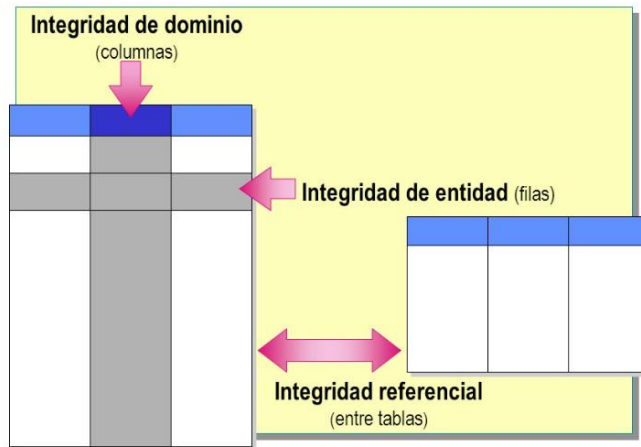
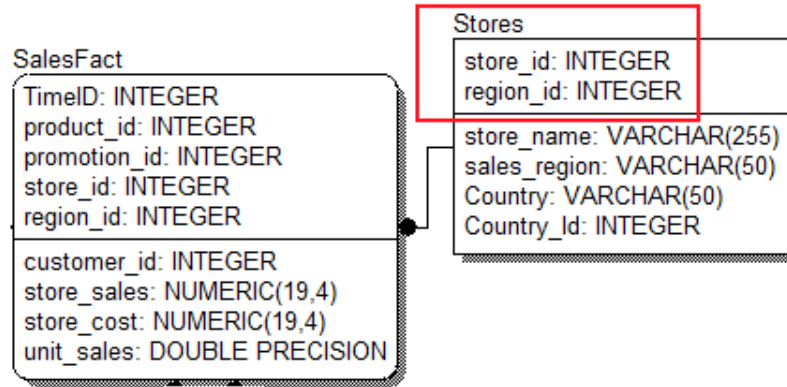


CODIGO	ALUMNO	PROMEDIO
458784	JORGE CACERES	11
487588	CATALINA RUIZ	12
457859	ROSA CASTAÑEDA	10
457897	JHONATAN SALVATORE	10
458978	MARIA DEL P. URIA	11
458979	EDWARD PATOW	12

Si el atributo (B) es funcionalmente dependiente de una llave compuesta (A), pero no de cualquier subconjunto de esa llave compuesta, el atributo (B) es total y funcionalmente dependiente de (A).

- **Dependencia funcional:** el valor de uno o más atributos determina el valor de uno o más atributos
 - **Determinante:** Atributo cuyo valor determina otro
 - **Dependiente:** Atributo cuyo valor está determinado por el otro atributo
- **Dependencia funcional completa:** Toda la colección de atributos en el determinante es necesaria para la relación

Tipos de llaves



- Clave compuesta: clave que se compone de más de un atributo
- Atributo de clave: atributo que forma parte de una clave
- Integridad de la entidad: condición en la que cada fila de la tabla tiene su propia identidad única
 - Todos los valores de la clave principal deben ser únicos
 - Ningún atributo de clave de la clave principal puede contener un valor nulo

Tipos de llaves



- Null: Ausencia de cualquier valor de datos que pueda representar:
 - Un valor de atributo desconocido
 - Un valor de atributo conocido, pero que falta,
 - Una condición inaplicable
- Integridad referencial: toda referencia a una instancia de entidad por otra instancia de entidad es válida

Llaves de base de datos relacional



Tipo de llave	Definición
Superllave	Un atributo(s) que de modo único identifica cada fila en una tabla.
Llave candidata	Una superllave mínima (irreductible); que no contiene un subconjunto de atributos que no sea en sí una superclave.
Llave primaria	Una llave candidata seleccionada para identificar de modo único todos los otros valores de atributo en cualquier fila. No puede contener entradas nulas.
Llave secundaria	Un atributo(s) que se usa estrictamente para fines de recuperación de datos
Llave foránea	Un atributo(s) de una tabla cuyos valores deben corresponderse con la llave primaria en otra tabla o ser nulos.

Ejemplo de una Base de Datos Relacional sencilla



Nombre de la tabla: **PRODUCT**

Nombre de la base de datos: **Ch03_SaleCo**

Llave primaria: **PROD_CODE**

Llave foránea: **VEND_CODE**

PROD_CODE	PROD_DESCRIPT	PROD_PRICE	PROD_ON_HAND	VEND_CODE
001278-AB	Claw hammer	12.95	23	232
123-21UUY	Houselite chain saw, 16-in. bar	189.99	4	235
QER-34256	Sledge hammer, 16-lb. head	18.63	6	231
SRE-657UG	Rat-tail file	2.99	15	232
ZZX/3245Q	Steel tape, 12-ft. length	6.79	8	235

enlace

Nombre de la tabla: **VENDOR**

Llave primaria: **VEND_CODE**

Llave foránea: ninguna

VEND_CODE	VEND_CONTACT	VEND_AREACODE	VEND_PHONE
230	Shelly K. Smithson	608	555-1234
231	James Johnson	615	123-4536
232	Annelise Crystall	608	224-2134
233	Candice Wallace	904	342-6567
234	Arthur Jones	615	123-3324
235	Henry Ortozo	615	899-3425

Reglas de integridad: Integridad de entidad



Requisito Todas las entradas de llave primaria son únicas y ninguna parte de una llave primaria puede ser nula.

Propósito Cada renglón tendrá una identidad única y los valores de llave foránea pueden referirse debidamente a valores de llave primaria.

Ejemplo Ninguna factura puede tener un número duplicado, ni puede ser nula. En pocas palabras, todas las facturas son identificadas de modo único por sus números de factura

Reglas de integridad: Integridad Referencial



Requisito

Una llave foránea puede tener ya sea una entrada nula, mientras no sea parte de la llave primaria de la tabla, o una entrada que sea igual al valor de la llave primaria en una tabla con la que está relacionada. (Todo valor no nulo de llave foránea debe referirse a un valor existente de clave primaria.)

Propósito

Es posible que un atributo NO tenga un valor correspondiente, pero será imposible que tenga una entrada inválida. La aplicación de la regla de integridad referencial hace imposible eliminar un renglón de una tabla cuya llave primaria tenga valores de llave foránea comparable y obligatorio en otra tabla.

Ejemplo

Un cliente podría no tener todavía un representante de ventas asignado (número), pero será imposible que tenga un representante de ventas no válido (número).

Formas de manejar nulls



- **Banderas:** Códigos especiales utilizados para indicar la ausencia de algún valor
- **Restricción NOT NULL:** se coloca en una columna para asegurarse de que cada fila de la tabla tiene un valor para esa columna
- **Restricción UNIQUE** - Restricción colocada en una columna para garantizar que no existan valores duplicados para esa columna

Valor de variable ficticia usado como indicador

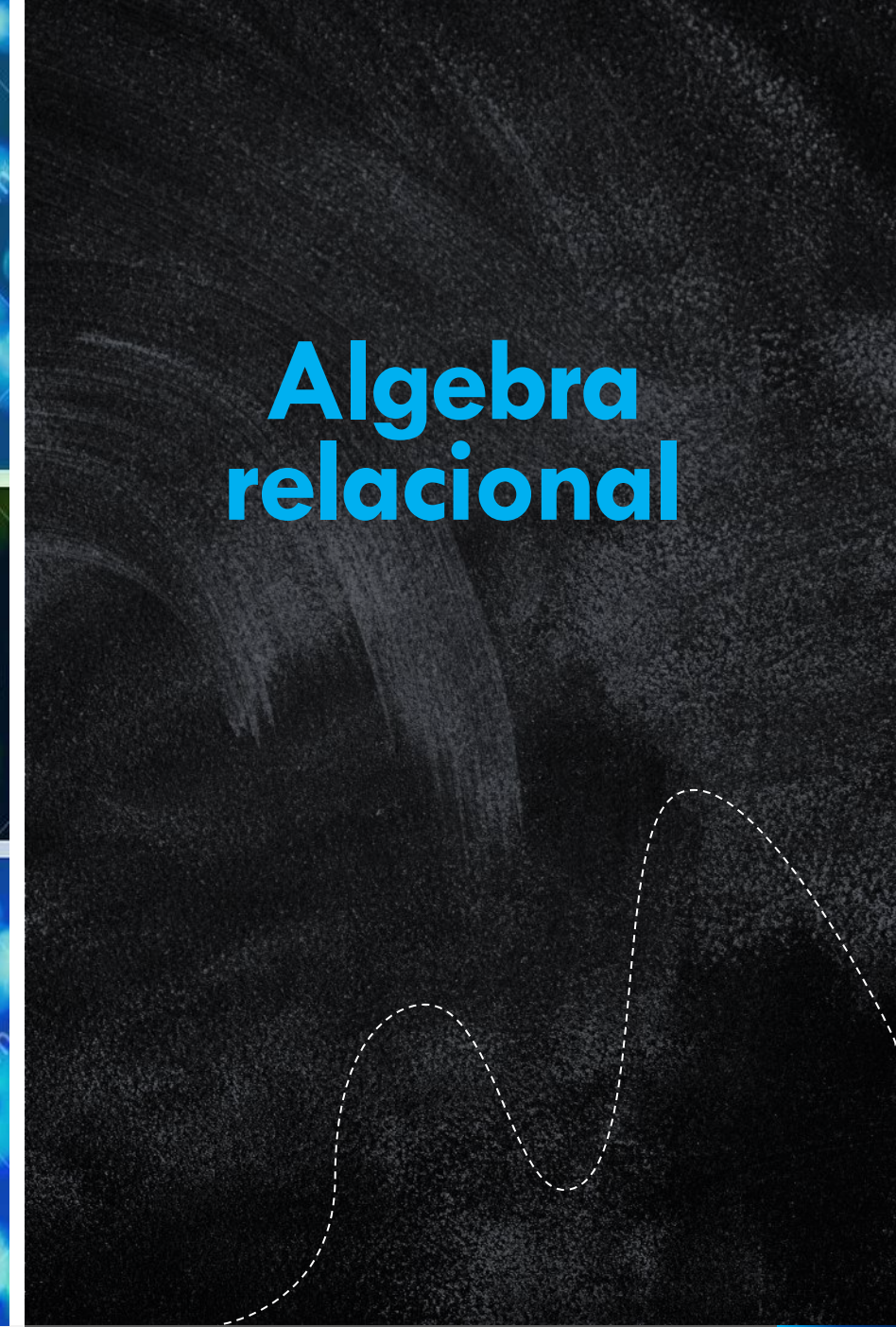
AGENT_CODE	AGENT_AREACODE	AGENT_PHONE	AGENT_LNAME	AGENT_YTD_SALES
-99	000	000-0000	Ninguno	\$0.00



RELATIONAL ALGEBRA



**Algebra
relacional**



Algebra relacional



- Forma teórica de manipular el contenido de la tabla utilizando operadores relacionales
- **Relvar:** Variable que mantiene una relación
 - El encabezado contiene los nombres de los atributos y el cuerpo contiene la relación
- Los operadores relacionales tienen la propiedad de cierre
 - **Cierre:** El uso de operadores de álgebra relacional en relaciones existentes produce nuevas relaciones

• Operadores

- El álgebra relacional define la forma teórica de manipular el contenido de una tabla usando los ocho operadores relacionales:
 - SELECT
 - JOIN
 - UNION
 - PRODUCT
 - PROJECT
 - INTERSECT
 - DIFFERENCE
 - DIVIDE

Select



- Da valores para todos los renglones de la tabla que satisfagan una condición dada. Se puede usar para hacer una lista de todos los valores de renglón, o puede dar sólo los valores de renglón que coincidan con un criterio especificado. Devuelve un subconjunto horizontal de una tabla.

Tabla original

P_CODE	P_DESCRIPT	PRICE
123456	Flashlight	5.26
123457	Lamp	25.15
123458	Box Fan	10.99
213345	9v battery	1.92
254467	100W bulb	1.47
311452	Powerdrill	34.99

SELECT ALL da

Nueva tabla

P_CODE	P_DESCRIPT	PRICE
123456	Flashlight	5.26
123457	Lamp	25.15
123458	Box Fan	10.99
213345	9v battery	1.92
254467	100W bulb	1.47
311452	Powerdrill	34.99

SELECT sólo PRICE menor a \$2.00 da

P_CODE	P_DESCRIPT	PRICE
213345	9v battery	1.92
254467	100W bulb	1.47

SELECT sólo P_CODE = 311452 da

P_CODE	P_DESCRIPT	PRICE
311452	Powerdrill	34.99

- Devuelve todos los valores para atributos seleccionados. En otras palabras, PROJECT da un subconjunto vertical de una tabla.

Tabla original

P_CODE	P_DESCRIPT	PRICE
123456	Flashlight	5.26
123457	Lamp	25.15
123458	Box Fan	10.99
213345	9v battery	1.92
254467	100W bulb	1.47
311452	Powerdrill	34.99

PROJECT PRICE da

Nueva tabla

PRICE
5.26
25.15
10.99
1.92
1.47
34.99

PROJECT P_DESCRIPT y PRICE da

P_DESCRIPT	PRICE
Flashlight	5.26
Lamp	25.15
Box Fan	10.99
9v battery	1.92
100W bulb	1.47
Powerdrill	34.99

PROJECT P_CODE y PRICE da

P_CODE	PRICE
123456	5.26
123457	25.15
123458	10.99
213345	1.92
254467	1.47
311452	34.99

Union



- Combina todos los renglones de dos tablas, excluyendo duplicados. Las tablas deben tener las mismas características de atributo (las columnas y dominios deben ser compatibles).
- Cuando dos o más tablas comparten el mismo número de columnas y cuando sus columnas correspondientes comparten los mismos (o compatibles) dominios, se dice que son compatibles en unión.

P_CODE	P_DESCRIPT	PRICE
123456	Flashlight	5.26
123457	Lamp	25.15
123458	Box Fan	10.99
213345	9v battery	1.92
254467	100W bulb	1.47
311452	Powerdrill	34.99

UNION

P_CODE	P_DESCRIPT	PRICE
345678	Microwave	160.00
345679	Dishwasher	500.00



da

P_CODE	P_DESCRIPT	PRICE
123456	Flashlight	5.26
123457	Lamp	25.15
123458	Box Fan	10.99
213345	9v battery	1.92
254467	100W bulb	1.47
311452	Powerdrill	34.99
345678	Microwave	160
345679	Dishwasher	500

INTERSECT



- Devuelve sólo los renglones que aparecen en ambas tablas. Por ejemplo, no se puede usar INTERSECT si uno de los atributos es numérico y otro es de caracteres.

STU_FNAME	STU_LNAME
George	Jones
Jane	Smith
Peter	Robinson
Franklin	Johnson
Martin	Lopez

INTERSECT

EMP_FNAME	EMP_LNAME
Franklin	Lopez
William	Turner
Franklin	Johnson
Susan	Rogers

da



STU_FNAME	STU_LNAME
Franklin	Johnson

DIFFERENCE



- Devuelve todos los renglones de una tabla que no se encuentren en la otra tabla
- Como en el caso de UNION, las tablas deben ser compatibles en unión para dar resultados válidos.
- Hay que tomar nota que la resta de la primera tabla de la segunda no es lo mismo que restar la segunda tabla de la primera.

STU_FNAME	STU_LNAME
George	Jones
Jane	Smith
Peter	Robinson
Franklin	Johnson
Martin	Lopez

DIFFERENCE

EMP_FNAME	EMP_LNAME
Franklin	Lopez
William	Turner
Franklin	Johnson
Susan	Rogers



STU_FNAME	STU_LNAME
George	Jones
Jane	Smith
Peter	Robinson
Martin	Lopez

PRODUCT



- Devuelve todos los pares posibles de renglones de dos tablas.
- Si una tabla tiene 6 renglones y la otra tiene 3, el PRODUCT da una lista compuesta de $6 \times 3 = 18$ renglones.

P_CODE	P_DESCRIPTOR	PRICE
123456	Flashlight	5.26
123457	Lamp	25.15
123458	Box Fan	10.99
213345	9v battery	1.92
254467	100W bulb	1.47
311452	Powerdrill	34.99

PRODUCT

STORE	aisle	shelf
23	W	5
24	K	9
25	Z	6

da



P_CODE	P_DESCRIPTOR	PRICE	STORE	aisle	shelf
123456	Flashlight	5.26	23	W	5
123456	Flashlight	5.26	24	K	9
123456	Flashlight	5.26	25	Z	6
123457	Lamp	25.15	23	W	5
123457	Lamp	25.15	24	K	9
123457	Lamp	25.15	25	Z	6
123458	Box Fan	10.99	23	W	5
123458	Box Fan	10.99	24	K	9
123458	Box Fan	10.99	25	Z	6
213345	9v battery	1.92	23	W	5
213345	9v battery	1.92	24	K	9
213345	9v battery	1.92	25	Z	6
311452	Powerdrill	34.99	23	W	5
311452	Powerdrill	34.99	24	K	9
311452	Powerdrill	34.99	25	Z	6
254467	100W bulb	1.47	23	W	5
254467	100W bulb	1.47	24	K	9
254467	100W bulb	1.47	25	Z	6

- Permite que se reúna información de dos o más tablas. JOIN es el poder real detrás de la base de datos relacional, ya que permite el uso de tablas independientes enlazadas por atributos comunes.
- Las tablas CUSTOMER y AGENT se usarán para ilustrar los ejemplos

Nombre de la tabla: CUSTOMER

CUS_CODE	CUS_LNAME	CUS_ZIP	AGENT_CODE
1132445	Walker	32145	231
1217782	Adares	32145	125
1312243	Rakowski	34129	167
1321242	Rodriguez	37134	125
1542311	Smithson	37134	421
1657399	Vanloo	32145	231

Nombre de la tabla: AGENT

AGENT_CODE	AGENT_PHONE
125	6152439887
167	6153426778
231	6152431124
333	9041234445

Join: Reunion natural



1) Primero, se crea un PRODUCT de las tablas

- Una reunión natural enlaza tablas al seleccionar sólo los renglones con valores comunes en su(s) atributo(s) común(es). Una reunión natural es el resultado de un proceso en tres etapas.
- El primero se muestra en esta diapositiva.

CUS_CODE	CUS_LNAME	CUS_ZIP	CUSTOMER.AGENT_CODE	AGENT.AGENT_CODE	AGENT_PHONE
1132445	Walker	32145	231	125	6152439887
1132445	Walker	32145	231	167	6153426778
1132445	Walker	32145	231	231	6152431124
1132445	Walker	32145	231	333	9041234445
1217782	Adares	32145	125	125	6152439887
1217782	Adares	32145	125	167	6153426778
1217782	Adares	32145	125	231	6152431124
1217782	Adares	32145	125	333	9041234445
1312243	Rakowski	34129	167	125	6152439887
1312243	Rakowski	34129	167	167	6153426778
1312243	Rakowski	34129	167	231	6152431124
1312243	Rakowski	34129	167	333	9041234445
1321242	Rodriguez	37134	125	125	6152439887
1321242	Rodriguez	37134	125	167	6153426778
1321242	Rodriguez	37134	125	231	6152431124
1321242	Rodriguez	37134	125	333	9041234445
1542311	Smithson	37134	421	125	6152439887
1542311	Smithson	37134	421	167	6153426778
1542311	Smithson	37134	421	231	6152431124
1542311	Smithson	37134	421	333	9041234445
1657399	Vanloo	32145	231	125	6152439887
1657399	Vanloo	32145	231	167	6153426778
1657399	Vanloo	32145	231	231	6152431124
1657399	Vanloo	32145	231	333	9041234445

Join: Reunion natural



2) Se realiza un **SELECT** a los registros con campos comunes

- En segundo término, se realiza un **SELECT** en la salida del paso a) para dar sólo los renglones para los cuales
- los valores **AGENT_CODE** son iguales. Las columnas comunes se conocen como columnas reunidas.

CUS_CODE	CUS_LNAME	CUS_ZIP	CUSTOMER.AGENT_CODE	AGENT.AGENT_CODE	AGENT_PHONE
1217782	Adares	32145	125	125	6152439887
1321242	Rodriguez	37134	125	125	6152439887
1312243	Rakowski	34129	167	167	6153426778
1132445	Walker	32145	231	231	6152431124
1657399	Vanloo	32145	231	231	6152431124

Join: Reunion natural



3) Se realiza un PROJECT mostrando los atributos necesarios

- Un PROJECT se realiza en los resultados del paso 2) para dar una sola copia de cada atributo, eliminando así las columnas duplicadas

CUS_CODE	CUS_LNAME	CUS_ZIP	AGENT_CODE	AGENT_PHONE
1217782	Adares	32145	125	6152439887
1321242	Rodriguez	37134	125	6152439887
1312243	Rakowski	34129	167	6153426778
1132445	Walker	32145	231	6152431124
1657399	Vanloo	32145	231	6152431124

Join: Equireunión - Reunión interior – Reunión exterior



- **Equireunión**. Enlaza tablas con base en una condición de igualdad que compara columnas especificadas de cada tabla.
- El resultado no elimina columnas duplicadas y la condición para reunir las tablas debe definirse explícitamente.
- La equireunión toma su nombre del operador de igualdad (=)
- Si se usa otro operador de comparación, la unión recibe el nombre de **reunión theta**.

- Es frecuente que cada una de las reuniones precedentes se clasifique como **reunión interior**. Una **reunión interior** es aquella que sólo retorna registros que se corresponden con las tablas que estén combinándose.

- En una **reunión exterior**, los pares que se corresponden se retienen y cualesquier valores que no tienen correspondencia en la otra tabla se dejarían nulos.
- Entrega también todos los registros con correspondencia de la reunión interior, además de los registros sin correspondencia de una de las tablas

Join: Reunión exterior izquierda y exterior derecha



- Una **reunión exterior izquierda** da todos los renglones de la tabla CUSTOMER, incluido los que no tengan un valor correspondiente en la tabla AGENT.
- Una **reunión exterior derecha** da todos los renglones de la tabla AGENT, incluyendo los que no tengan valores correspondientes en la tabla CUSTOMER.

CUS_CODE	CUS_LNAME	CUS_ZIP	AGENT_CODE	AGENT_PHONE
1217782	Adares	32145	125	6152439887
1321242	Rodriguez	37134	125	6152439887
1312243	Rakowski	34129	167	6153426778
1132445	Walker	32145	231	6152431124
1657399	Vanloo	32145	231	6152431124
1542311	Smithson	37134	421	

CUS_CODE	CUS_LNAME	CUS_ZIP	AGENT_CODE	AGENT_PHONE
1217782	Adares	32145	125	6152439887
1321242	Rodriguez	37134	125	6152439887
1312243	Rakowski	34129	167	6153426778
1132445	Walker	32145	231	6152431124
1657399	Vanloo	32145	231	6152431124
			333	9041234445

Diccionario de datos y catálogo del sistema

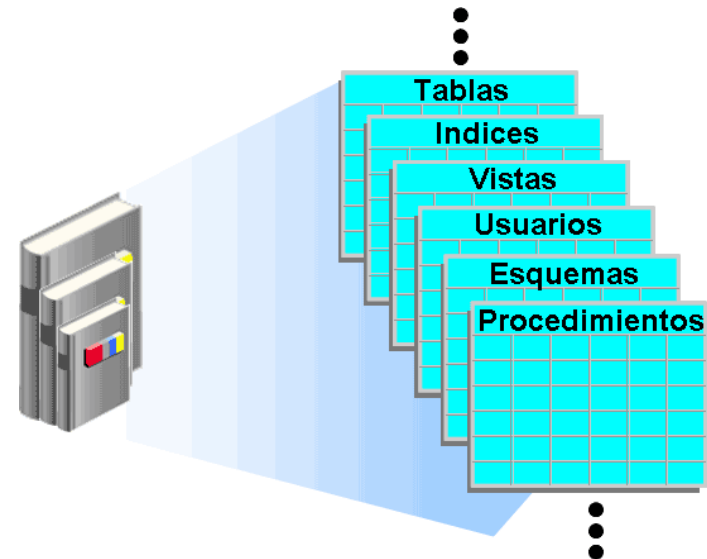


- **Diccionario de datos:** descripción de todas las tablas de la base de datos creadas por el usuario y el diseñador
- **Catálogo del sistema:** diccionario de datos del sistema que describe todos los objetos de la base de datos
- Los homónimos y sinónimos deben evitarse para disminuir la confusión
 - Homónimo: El mismo nombre se utiliza para etiquetar diferentes atributos
 - Sinónimo: Se utilizan diferentes nombres para describir el mismo atributo

Muestra de un diccionario de datos

NOMBRE DE TABLA	NOMBRE DE ATRIBUTO	CONTENIDO	TIPO	FORMATO	INTERVALO	REQUERIDA	PK O FK	TABLA DE REFERENCIA FK
CUSTOMER	CUS_CODE	Código de cuenta de cliente	CHAR(5)	99999	10000-99999	Y	FK	
	CUS_LNAME	Apellido de cliente	VARCHAR(20)	Xxxxxxxxx		Y		
	CUS_FNAME	Nombre de cliente	VARCHAR(20)	Xxxxxxxxx		Y		
	CUS_INITIAL	Inicial de cliente	CHAR(1)	X				
	CUS_RENEW_DATE	Fecha de renovación de seguro de cliente	DATE	dd-mm-aaaa				
AGENT	AGENT_CODE	Código de agente	CHAR(3)	999			FK	AGENT_CODE
	AGENT_CODE	Código de agente	CHAR(3)	999		Y		
	AGENT_AREACODE	Código postal de agente	CHAR(3)	999		Y		
	AGENT_PHONE	Número telefónico de agente	CHAR(8)	999-9999		Y		
	AGENT_LNAME	Apellido de agente	VARCHAR(20)	Xxxxxxxxx		Y		
	AGENT_YTD_SLS	Ventas de agente de un año a la fecha	NUMBER(9,2)	9,999,999.99		Y		

FK = Llave foránea
PK = Llave primaria
CHAR = Datos de longitud fija de caracteres
VARCHAR = Datos de longitud variable de caracteres
NUMBER = Datos numéricos (NUMBER(9,2)) se usan para especificar números con dos lugares decimales y hasta nueve dígitos, incluido lugares decimales.



Relaciones dentro de la BD Relacional



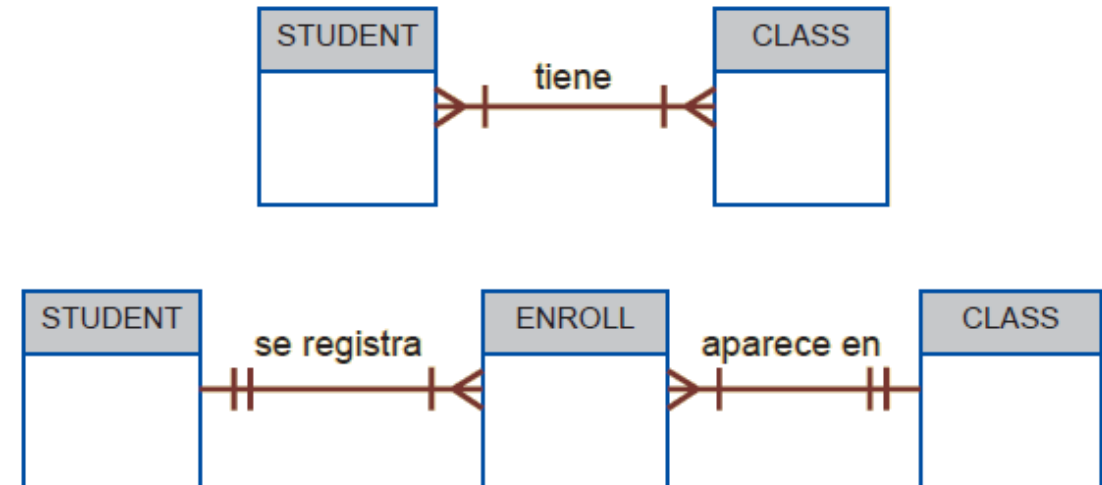
- Relación 1:M - Norma para bases de datos relacionales
- Relación 1:1 - Una entidad puede estar relacionada con sólo otra entidad y viceversa
- Relación de muchos a varios (M:N): se implementa mediante la creación de una nueva entidad en relaciones 1:M con las entidades originales.
- Entidad compuesta (Bridge o entidad asociativa): Ayuda a evitar problemas inherentes a las relaciones M:N, incluye las claves primarias de las tablas a enlazar

Nombre de la tabla: PAINTER
Llave primaria: PAINTER_NUM
Llave foránea: ninguna

PAINTER_NUM	PAINTER_LNAME	PAINTER_FNAME	PAINTER_INITIAL
123	Ross	Georgette	P
126	Ittero	Julio	G

Nombre de la tabla: PAINTING
Clave primaria: PAINTING_NUM
Clave foránea: PAINTER_NUM

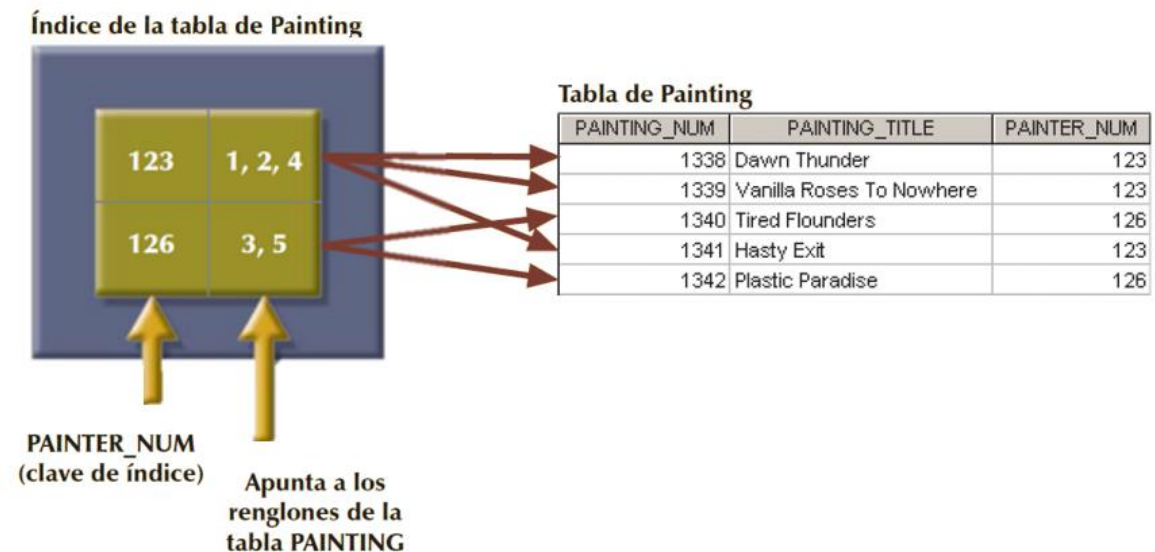
PAINTING_NUM	PAINTING_TITLE	PAINTER_NUM
1338	Dawn Thunder	123
1339	Vanilla Roses To Nowhere	123
1340	Tired Flounders	126
1341	Hasty Exit	123
1342	Plastic Paradise	126



Índices



- Disposición ordenada para acceder lógicamente a las filas de una tabla
- Clave de índice: punto de referencia del índice que conduce a la ubicación de datos identificada por la clave.
- Índice único: la clave de índice solo puede tener un valor de puntero asociado
- Cada índice está asociado a una sola tabla



**Recordemos lo
aprendido**



Recordemos lo aprendido

- ¿Qué características tiene una tabla relacional?
- ¿Qué es una llave? ¿Para qué se usa?
- ¿Qué entiende por determinación?
- De un ejemplo de dependencia funcional
- ¿Qué entiende por integridad de la entidad?
- ¿Para qué sirve la integridad referencial? ¿De qué manera sería útil en un caso real?
- ¿Qué es null?
- ¿Qué tipos de llaves conoce? ¿Cuál considera de importancia en el modelo relacional?



Siga recordando!!!



- ¿De qué maneras puede manejar los nulls en una base de datos?
- En el álgebra relacional en qué se diferencia el select del Project
- ¿Existe diferencia entre el Join y el unión?
- Describa el proceso seguido para hacer un join
- ¿Qué es un diccionario de datos?
- ¿Cómo transformamos un modelo ER a un modelo relacional?
- ¿Para qué sirven los índices?



Aplicando lo aprendido

Ejercicio de aplicación: Dada la siguiente BD



Table name: EMPLOYEE

Database name: Ch03_StoreCo

EMP_CODE	EMP_TITLE	EMP_LNAME	EMP_FNAME	EMP_INITIAL	EMP_DOB	STORE_CODE
1	Mr.	Williamson	John	vW	21-May-64	3
2	Ms.	Ratula	Nancy		09-Feb-69	2
3	Ms.	Greenboro	Lottie	R	02-Oct-61	4
4	Mrs.	Rumpersfro	Jennie	S	01-Jun-71	5
5	Mr.	Smith	Robert	L	23-Nov-59	3
6	Mr.	Renselaer	Cary	A	25-Dec-65	1
7	Mr.	Ogallo	Roberto	S	31-Jul-62	3
8	Ms.	Johnsson	Elizabeth	I	10-Sep-68	1
9	Mr.	Eindsmar	Jack	vW	19-Apr-55	2
10	Mrs.	Jones	Rose	R	06-Mar-66	4
11	Mr.	Broderick	Tom		21-Oct-72	3
12	Mr.	Washington	Alan	Y	08-Sep-74	2
13	Mr.	Smith	Peter	N	25-Aug-64	3
14	Ms.	Smith	Sherry	H	25-May-66	4
15	Mr.	Olenko	Howard	U	24-May-64	5
16	Mr.	Archialo	Barry	V	03-Sep-60	5
17	Ms.	Grimaldo	Jeanine	K	12-Nov-70	4
18	Mr.	Rosenberg	Andrew	D	24-Jan-71	4
19	Mr.	Rosten	Peter	F	03-Oct-68	4
20	Mr.	Mckee	Robert	S	06-Mar-70	1
21	Ms.	Baumann	Jennifer	A	11-Dec-74	3

Table name: STORE

STORE_CODE	STORE_NAME	STORE_YTD_SALES	REGION_CODE	EMP_CODE
1	Access Junction	1003455.76	2	8
2	Database Corner	1421987.39	2	12
3	Tuple Charge	986783.22	1	7
4	Attribute Alley	944568.56	2	3
5	Primary Key Point	2930098.45	1	15

Table name: REGION

REGION_CODE	REGION_DESCRIPT
1	East
2	vWest

Resuelva



1. Para cada tabla, identifique la llave primaria y la(s) llave(s) foránea(s). Si una tabla no tiene una llave foránea, escriba ninguna en el espacio asignado.

TABLA	LLAVE PRIMARIA	LLAVE(S) FORÁNEA(S)
EMPLOYEE		
STORE		
REGION		

2. ¿Las tablas exhiben integridad de entidad? Conteste sí o no y luego explique su respuesta.

TABLA	INTEGRIDAD DE ENTIDAD	LLAVE(S) FORÁNEA(S)
EMPLOYEE		
STORE		
REGION		

3. ¿Las tablas exhiben integridad referencial? Conteste sí o no y luego explique su respuesta. Escriba NA (no aplicable) si la tabla no tiene una clave foránea.

- 4. Describa el o los tipos de relaciones entre STORE y REGION.
- 5. Genere el ERD para mostrar la relación entre STORE y REGION.
- 6. Genere el diagrama relacional para mostrar la relación entre EMPLOYEE, STORE y REGION.

TABLA	INTEGRIDAD REFERENCIAL	LLAVE(S) FORÁNEA(S)
EMPLOYEE		
STORE		
REGION		

