**INSERT**

Podemos insertar en una tabla el resultado de una consulta sobre otra tabla. En este caso normalmente se insertarán varias filas con una sola sentencia. Utilizaremos el siguiente formato:

**INSERT INTO NombreTabla [( NombreColumna [,NombreColumna...] ) ]**

**SELECT FormatoSelect**

*Notación: la lista de columnas en las que insertamos va es opcional, por lo que va entre*

*corchetes.*

En el formato anterior podemos destacar:

· La **lista de columnas es opcional** pero deberá especificarse cuando las columnas que devuelve la consulta no coinciden en número o en orden con las columnas de la tabla destino.

· La **consulta puede ser cualquier comando de selección** válido siempre que exista una correspondencia entre las columnas devueltas y las columnas de la tabla destino, o la lista de columnas.

Cread las tablas PedidosHistorico y DetallePedidos\_historico con los campos y las restricciones siguientes que se corresponden con los de Pedidos y DetallePedidos.

* Para PedidosHistorico añadiremos un campo, FechaIntro, donde introduciremos la fecha actual.
* En el caso de DetallePedidos\_historico, la clave ajena será hacia PedidosHistorico.

CREATE TABLE PedidosHistorico (

CodigoPedido integer NOT NULL,

FechaPedido date NOT NULL,

FechaEsperada date NOT NULL,

FechaEntrega date DEFAULT NULL,

CodigoCliente integer NOT NULL,

..FechaIntro date,

PRIMARY KEY (CodigoPedido),

CONSTRAINT pedHist\_Cliente FOREIGN KEY (CodigoCliente) REFERENCES Clientes (CodigoCliente)

) engine=innodb;

CREATE TABLE DetallePedidosHistorico (

CodigoPedido integer NOT NULL,

CodigoProducto varchar(15) NOT NULL,

Cantidad integer NOT NULL,

PrecioUnidad numeric(15,2) NOT NULL,

NumeroLinea smallint NOT NULL,

PRIMARY KEY (CodigoPedido,CodigoProducto),

CONSTRAINT HDetallePedidos\_PedidoFK FOREIGN KEY (CodigoPedido) REFERENCES PedidosHistorico (CodigoPedido),

CONSTRAINT HDetallePedidos\_ProductoFK FOREIGN KEY (CodigoProducto) REFERENCES Productos (CodigoProducto)

)engine=innodb;

1. Introducid en la tabla Empleados 3 empleados nuevos. Recordad que tendréis que asignarlos a una OFICINA válida.

Si miráis el script Jardinería, la tabla empleados tiene la siguiente estructura:

CREATE TABLE Empleados (

CodigoEmpleado integer NOT NULL,

Nombre varchar(50) NOT NULL,

Apellido1 varchar(50) NOT NULL,

Apellido2 varchar(50) DEFAULT NULL,

Extension varchar(10) NOT NULL,

Email varchar(100) NOT NULL,

CodigoOficina varchar(10) NOT NULL,

CodigoJefe integer DEFAULT NULL,

Puesto varchar(50) DEFAULT NULL,

PRIMARY KEY (CodigoEmpleado),

CONSTRAINT Empleados\_OficinasFK FOREIGN KEY (CodigoOficina) REFERENCES Oficinas (CodigoOficina),

CONSTRAINT Empleados\_EmpleadosFK FOREIGN KEY (CodigoJefe) REFERENCES Empleados (CodigoEmpleado)

) engine=innodb;

Por lo tanto, para introducir un empleado tendremos obligatoriamente que cumplimentar todos los campos obligatorios con valores válidos y podemos cumplimentar o no los opcionales: Apellido2, CodigoJefe y Puesto. Vamos a rellenar solo los obligatorios

Si ejecutáis:

select max(CodigoEmpleado) from Empleados

Veréis que puedo poner CodigoEmpleado a partir del 32

Para saber las Oficinas válidas:

select CodigoOficina from Oficinas

Así una posible solución sería:

INSERT INTO Empleados values (32,'Lucía','Gimeno',DEFAULT,'3897','lucia@jardineria.es','TAL-ES',DEFAULT,DEFAULT), (33,'Raúl','Pérez',DEFAULT,'3897','raul@jardineria.es','TAL-ES',DEFAULT,DEFAULT),

(34,'María','García',DEFAULT,'3897','maria@jardineria.es','TAL-ES',DEFAULT,DEFAULT)

DEFAULT podría haber sido NULL o cualquier otro valor válido.

1. Introducid en la tabla clientes 3 clientes que estén asignados a cada uno de los nuevos empleados creados en el ejercicio anterior.

Para introducir en Clientes hubiéramos hecho un proceso análogo teniendo en cuenta que el CodigoEmpleadoRepVentas tiene que ser 32, 33 y 34

Así, por ejemplo:

INSERT INTO Clientes VALUES

(39,'EMPRESA39',DEFAULT,DEFAULT,'393939393','5556901746','False Street 52 2 A',DEFAULT,'San Francisco', DEFAULT, DEFAULT, DEFAULT,**32**, DEFAULT),

(40,'EMPRESA40',DEFAULT,DEFAULT,'393939393','5556901746','False Street 52 2 A',DEFAULT,'San Francisco', DEFAULT, DEFAULT, DEFAULT,**33**, DEFAULT),

(41,'EMPRESA41',DEFAULT,DEFAULT,'393939393','5556901746','False Street 52 2 A',DEFAULT,'San Francisco', DEFAULT, DEFAULT, DEFAULT,**34**, DEFAULT)

1. Introducid en la tabla Pedidos\_historico los pedidos que ya se han servido (tienen fecha de entrega)

Pedidos\_historico no tiene los campos Estado ni Comentarios y Pedidos sí. Por otra parte, tiene el campo FechaIntro que Pedidos no tiene. Vamos a rellenar todos los campos. El campo FechaIntro con la fecha actual.

INSERT INTO PedidosHistorico

SELECT CodigoPedido, FechaPedido, FechaEsperada, FechaEntrega,CodigoCliente, **CURDATE()**

FROM Pedidos

WHERE FechaEntrega IS NOT NULL

1. Introducid en la tabla DetallePedidos\_historico las filas de DetallePedidos relacionadas con las filas de Pedidos\_historico.

Cualquiera de estas opciones sería válida:

INSERT INTO DetallePedidosHistorico

SELECT \* FROM DetallePedidos

WHERE CodigoPedido IN(SELECT CodigoPedido FROM PedidosHistorico)

INSERT INTO DetallePedidosHistorico

SELECT \* FROM DetallePedidos

WHERE CodigoPedido IN(SELECT CodigoPedido FROM Pedidos

WHERE FechaEntrega IS NOT NULL)

INSERT INTO DetallePedidosHistorico

SELECT d.\*

FROM DetallePedidos d INNER JOIN PedidosHistorico p ON d.CodigoPedido=p.CodigoPedido

INSERT INTO DetallePedidosHistorico

SELECT d.\*

FROM DetallePedidos d INNER JOIN Pedidos p ON d.CodigoPedido=p.CodigoPedido

WHERE FechaEntrega IS NOT NULL)