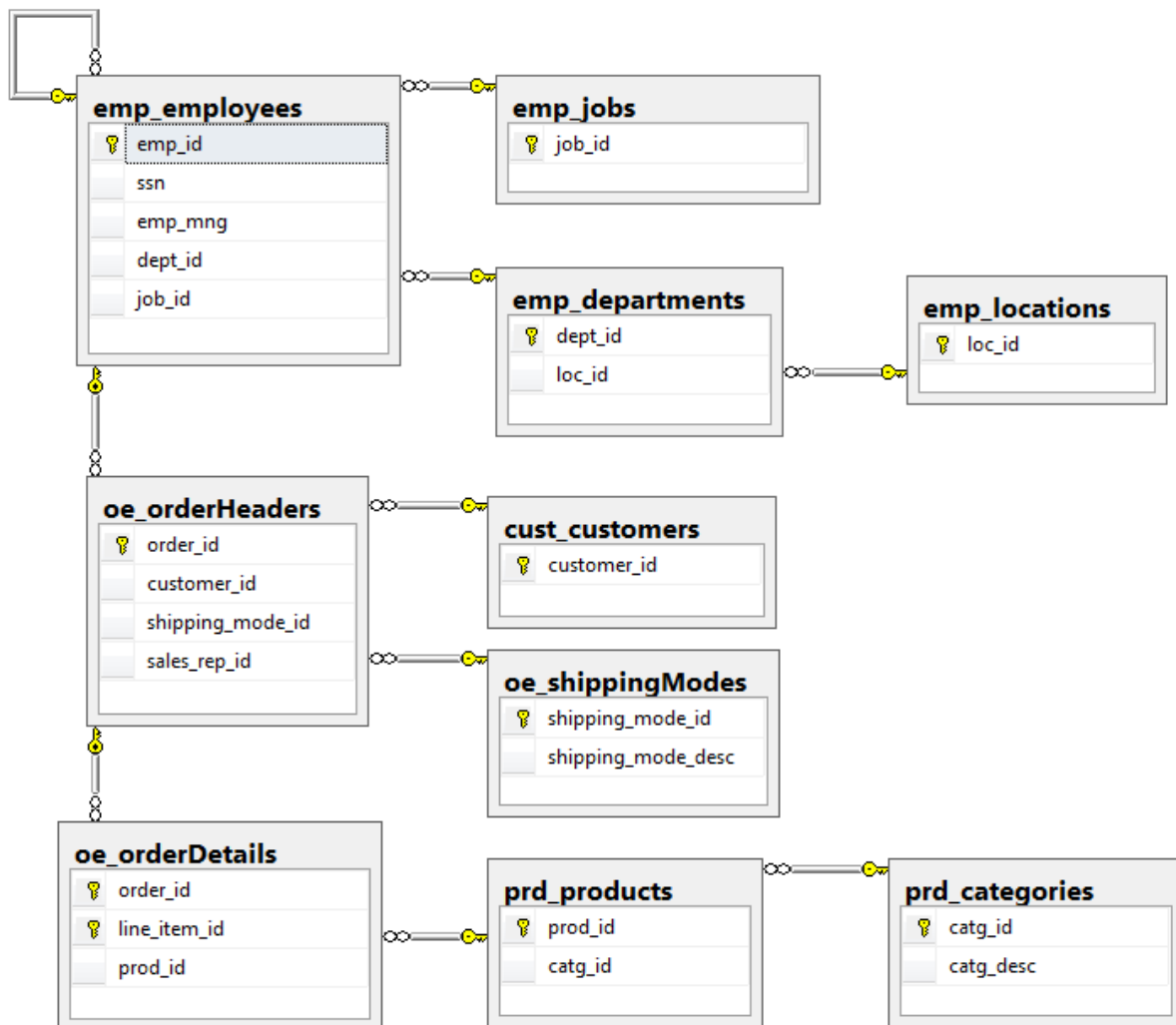


AltgeldMart Design

Many of the demos will use the following sets of tables belong to AltgeldMart. This discusses the business rules and the relationships for this database. You should read the Create Table statements for the details. Those statements are in the supplied scripts.

Our company (AltgeldMart) stores data about employees and the products sold and customer orders. These tables store only some of the data that would be needed by a company.

This first diagram shows the relations between the tables. I have limited the display to show only the table names and the keys (pk and fk) attributes.



PRODUCT

Altgeld_Mart sells a variety of products.

prd_categories: Each product is classified as being in a category such as Housewares, Pet Supplies, and Sporting Goods. This table is used to limit the values that can be entered as a category for a product. The table has a short value for each category as the primary key and a longer descriptive value. The descriptive values are unique. For example we have the rows:

catg_id	catg_desc
APL	APPLIANCES
BK	BOOKS

prd_products: A product has a name that is fairly short and a description, which could be longer. A product has only a single category. Products also have a list price, but we do not always sell items at their list price.

prd_products			
	Column Name	Data Type	Allow Nulls
?	prod_id	int	<input type="checkbox"/>
	prod_name	varchar(25)	<input type="checkbox"/>
	prod_desc	varchar(50)	<input checked="" type="checkbox"/>
	prod_list_price	decimal(6, 2)	<input type="checkbox"/>
	catg_id	varchar(6)	<input type="checkbox"/>
			<input type="checkbox"/>

prd_categories			
	Column Name	Data Type	Allow Nulls
?	catg_id	varchar(6)	<input type="checkbox"/>
	catg_desc	varchar(25)	<input type="checkbox"/>
			<input type="checkbox"/>

CUSTOMER

cust_customers: For customers, we are storing only the customer name and credit limit to keep the tables smaller.

cust_customers			
	Data Type	Column Name	Allow Nulls
?	int	customer_id	<input type="checkbox"/>
	varchar(25)	customer_name_last	<input type="checkbox"/>
	varchar(25)	customer_name_first	<input checked="" type="checkbox"/>
	int	customer_credit_limit	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

ORDERENTRY

These are the tables associated with the order entry component.

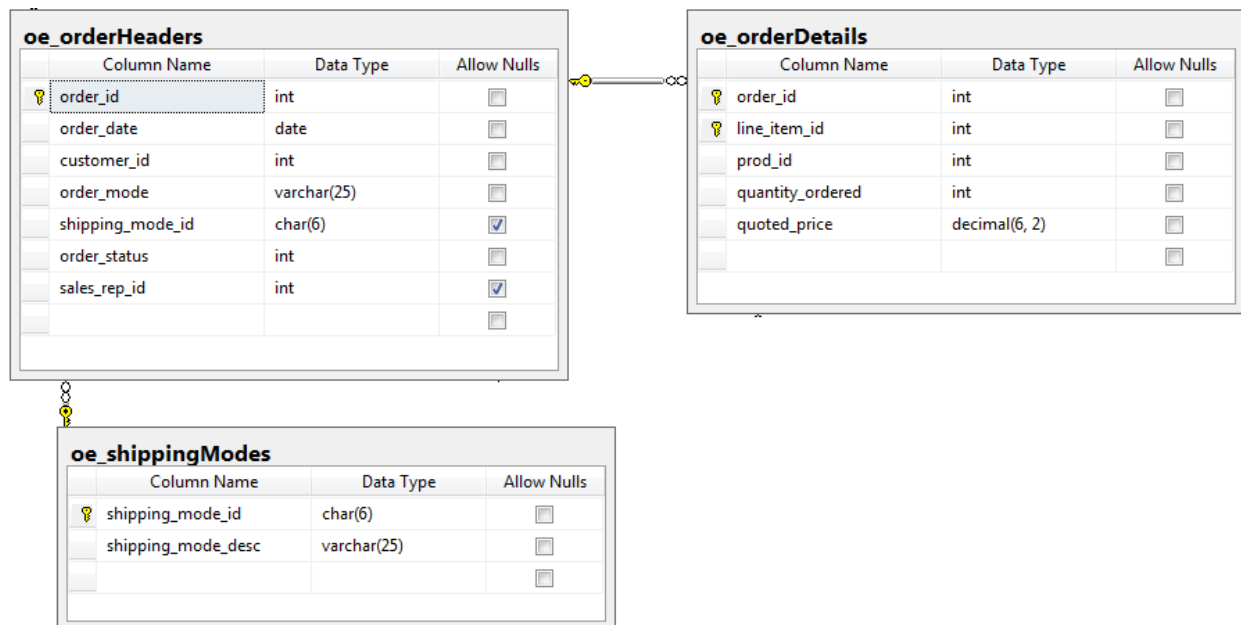
oe_orderHeaders: Each sales order belongs to a single customer(customer_id) ; an order has an order_date, an order mode, a shipping mode and a numeric order status. For some sales, we know the sales rep (employee) who handled the order.

oe_orderDetails: The order details tables includes the product being ordered, the quantity and the price at which the item was actually sold. By putting the actual sales price here, we could sell a product for less or more than its list price. If the list price changes, the actual price to the customer is not changed. The product id references the products table in the previous diagram.

The price in the order_details table is the price per item. For example, on order 114, the customer bought 5 mini freezers for \$125.00 each. For this order, the customer is charged \$625.00 (5 * 125) plus any tax and shipping.

The orderDetails table includes the order_id to link to the orderHeaders table. It also has a line_item_id, which is usually numbered 1,2,3, etc. Together, order_id and line_item_id make up the primary key for the orderDetails table.

oe_shippingModes: this table is similar to the product categories table. It lists the various types of shipping modes we use and each order is limited to one of these shipping modes.



EMPLOYEE

emp_employees: We store an employee's name. An employee has a single job title and is assigned to a single department. An employee may have a manager who is also an employee. This results in a relationship from the employees table back to itself. A relationship that a table has with itself is sometimes called a pig's ear. We store the employee's hire date and current salary.

emp_jobs: Each employee has a single job title and there is a table of the valid job titles. This also contains a minimum and maximum salary for that job

emp_departments: For each department, we store a department name and a single location.

emp_locations: The locations for the company sites are stored in a table of locations; we store the address information for each location.

