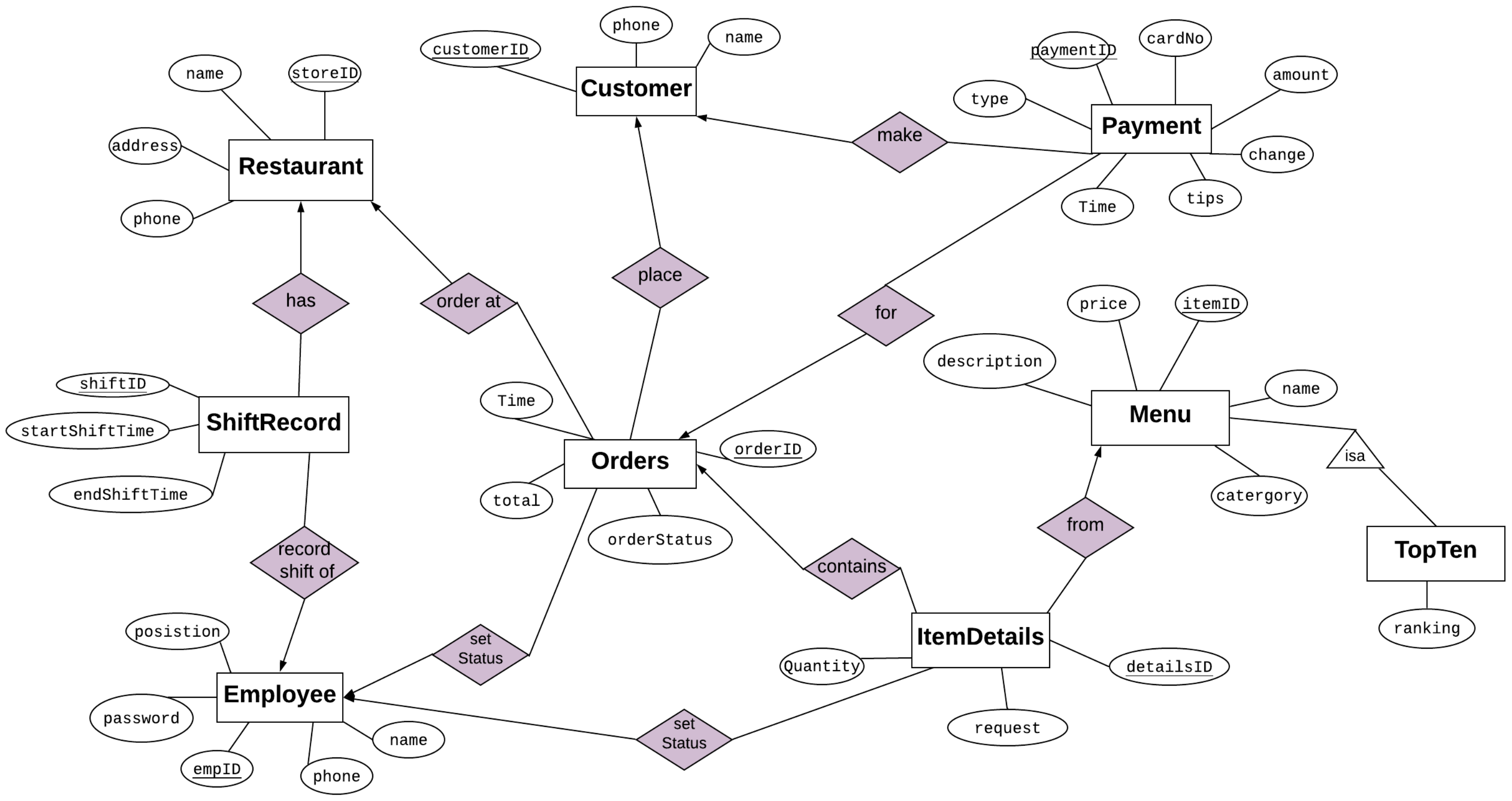
Database Design
Self-Serve Dining System

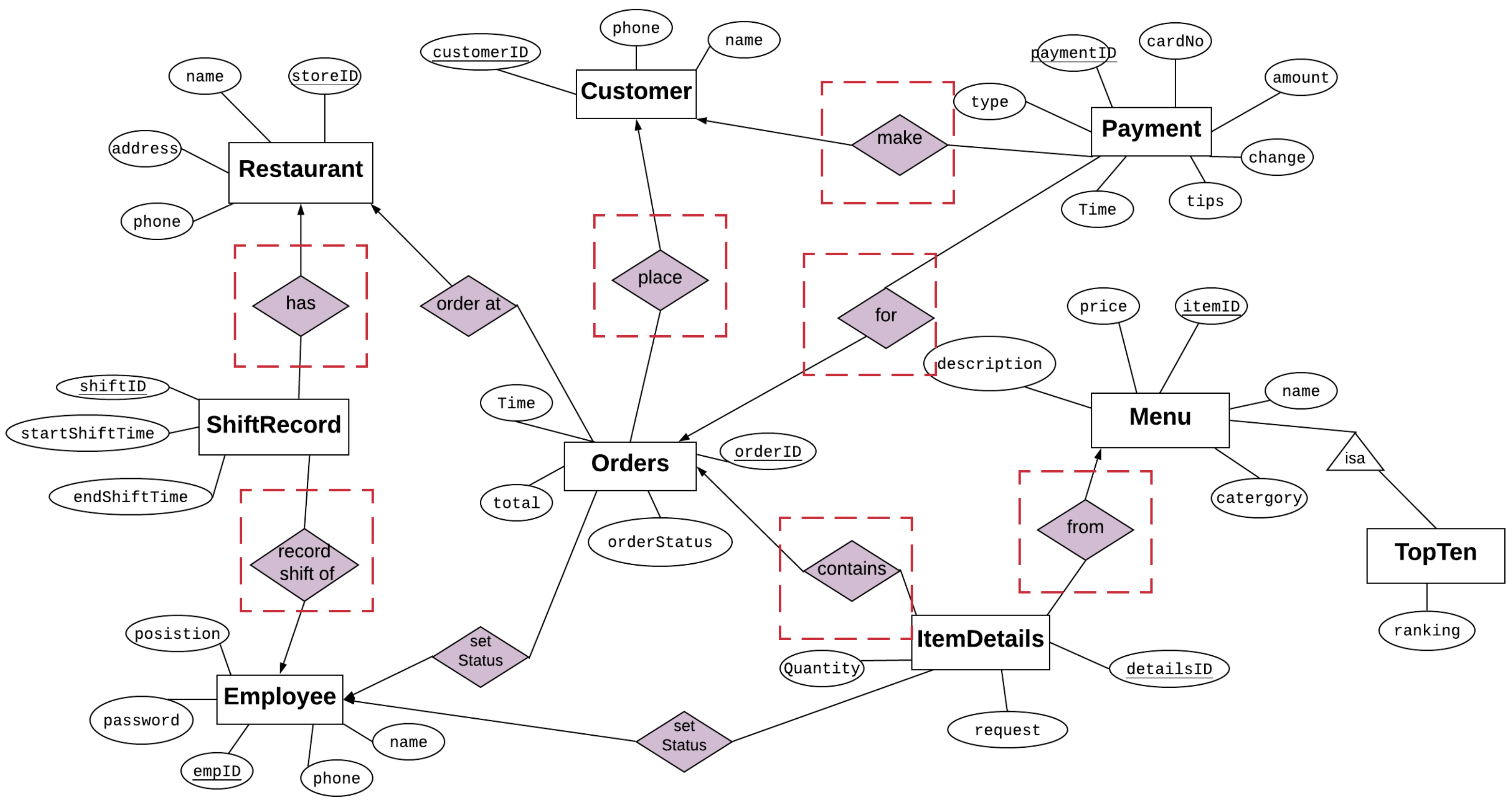
CS157A
Professor: Dr. Mike Wu
Oct 7, 2019

Group#8
Tao Sheng
Mavis Tsz Ching Tsoi
Angel Nguyen

Self-order Dining System E/R Diagram



Self-order Dining System E/R Diagram



=combining relations in database Schema

Database Schema

Restaurant (storeID, name, address, phone)

Employee (empID, name, phone, position, password)

ShiftRecord (shiftID, startShiftTime, endShiftTime, storeID, empID)

Customer (customerID, name, phone)

Menu (itemID, name, description, category, price)

TopTen (itemID, ranking)

Orders (orderID, time, total, orderStatus, customerID)

ItemDetails (detailsID, orderID, itemID, Quantity, request)

Payment (paymentID, customerID, orderID, time, type, cardNo, amount, change, tips)

orderAt (orderID, storeID)

setOrderStatus (empID, orderID)

setItemDetailsStatus (empID, detailsID)

Description

In the E/R diagram, 7 of the relationships had combined with the “many” side of entities for easy access. Since the relationships will be access a lot. Including the primary key of “one” entity to “many” entity can reduce database access. The rest relationships has less significant use for the related entity. Thus they are stored as a separated table.

Relation Restaurant represents entity “Restaurant”. It records the location of the restaurant in case there are multiple locations. storeID is the primary, it also contains name, address and phone number as attributes of a tuple.

Relation Employee represents entity “Employee”. It is a table records a list of all employee in the restaurant who have access to the system. empID is the primary key. Password attribute contains the password of employee to use for identification. Position will determine access level the system (e.g. admin/regular staff).

Relation ShiftRecord represents entity “ShiftRecord”, relationship “has” and relationship “record shift of”. It records the workhours of employee. This relation is combining the many to 1 relationship to relation Employee and the many to 1 relationship to relation Restaurant. One restaurant has multiple shift record. One employee has multiple shift record. A shift record records exact one employee at exact one restaurant. storeID and empID as foreign key are included to combine relationships.

Relation Customer represent entity “customer”. record the identity of customer. It contains a unique ID, name and phone no. of customer.

Relation Menu represents entity “Menu”. It records all food item sell in the restaurant. It contains itemID as primary, name, price, category and description of food as attributes. Category refers to appetizer, main course, drinks, etc.

Relation TopTen is the subclass of relation Menu. It records the top ten order item from the menu. It contains ranking# as a attribute.

Relation Orders represents entity “Orders”. It records the food order place by customers. It combines the many to 1 relationship to relation Customer. One customer makes multiple orders and an order is placed by exactly one customer. customerID as foreign key are included to combine this relationship.

Relation ItemDetails represents entity “ItemDetails”, relationship “contains” and relationship “from”. It records the content of each order from an item in menu. This relation is combining the many to 1 relationship to relation Orders and the many to 1 relationship to relation Menu. One order has multiple ItemDetails. One menu item has multiple ItemDetails from different orders. An ItemDetails records exact one menu item ordered in exact one Orders. orderID and itemID as foreign key are included to combine relationships. Request is the attribute store orders special request such as less salt in the food order.

Relation payment represents entity “Payment”, relationship “make” and relationship “for”. record the payment details from customers for an order. This relation is combining the many to 1 relationship to relation Orders and the many to 1 relationship to relation Customer. One order can have multiple payments (partial payments). One customer can make multiple payments from different orders. A payment records exact one customer’s payment for exact one order. orderID and customerID as foreign key are included to combine these relationships. type is the attribute store types of payment such as cash or visa.

Relation orderAt represents relationship “order at”. It contains primary key of restaurant relation and orders relation. storeID and orderID is the primary key of this relation. It Records the restaurant location of each order.

Relation setOrderStatus represents relationship “set status”. It contains primary key of employee relation and orders relation. empID and orderID is the primary key of this relation. It Records the which customer service employee set the entire order complete.

Relation setItemDetailsStatus represents relationship “set status”. It contains primary key of employee relation and ItemDetails relation. empID and detailsID is the primary key of this relation. It Records the which kitchen employee prepare an item from an order.