25) Нарисуйте общую схему модификаций ER-модели Чена. Какие концепции добавлялись и удалялись на каждом из этапов этих модификаций?



1. ER-модель была пополнена понятиями «специализация» и «категоризация» (для представления отношений обобщения между множествами сущностей, и, таким образом, образована расширенная ER-модель (Enhanced Entity-Relationship Model – EER-модель)).
2. В ER-модели Баркера удалились такие формы как:

* множества связей степени больше двух,
* атрибуты множеств связей,
* многозначные атрибуты,
* атрибутные отображения в Декартово произведение множеств значений.

1. В нотация IDEF1X удалились такие формы как:

* бинарные множества связей типа *M*:*N* (Их надо представлять в виде дополнительного множества сущностей и пары множеств связей типа 1:*M).*
* внутрисистемные указатели сущностей (Теперь надо было в явной форме оперировать первичными и внешними ключами Теперь сам проектировщик должен заботиться о создании суррогатных ключей).

26) В каких случаях используются специализация и категоризация?

Категоризация – моделирование одного подкласса со связью, которая охватывает несколько разных суперклассов (или моделирование сущности путем включения в нее характеристик других сущностей (множественное наследование)).

Специализация – процесс деления субъкта более высокого уровня на субеъкты нижнего уровня (или процесс увеличения различий между отдельными сущностями за счет выделения их отличительных характеристик).

49) Что такое внешний ключ? Должен ли он обладать свойством уникальности? Для чего и как он используется?

Внешний ключ – атрибут связи с другой сущностью, который обычно соответсвует первичному ключу другой сущности. Уникальность не обязательна. Используется для объединения двух таблиц. Внешний ключ - это столбец (или комбинация столбцов), значения которого(ых) соответсвуют первичному ключу в другой таблице.

85) Укажите различные случаи применения агрегатных функций в списке выборки в зависимости от вида табличного выражения. Приведите примеры.

Наиболее часто применяются агрегатные функции SQL SUM, MIN, MAX, AVG, COUNT. Следует различать два случая применения агрегатных функций:

1. Агрегатные функции исользуются сами по себе и возвращают одно результирующее значение.

* Функция SQL SUM - возвращает сумму значений столбца таблицы базы данных. Она может применяться только к столбцам, значениями которых являются числа. (**Пример**: SELECT SUM(З/П) FROM(СОТРУДНИК) WHERE(Job='Clerk'). Почитаем зарплату сотрудников с должностью Clerk).
* Функция SQL MIN - также действует в отношении столбцов, значениями которых являются числа и возвращает минимальное среди всех значений столбца. (**Пример**: SELECT MIN(З/П) FROM(СОТРУДНИК) WHERE(Dept=42). Узнаем минимальную зарплату сотрудников отдела с номером 42).
* Функция SQL MAX - применяется, когда требуется определить максимальное значение среди всех значений столбца. (**Пример**: SELECT MAX(З/П) FROM(СОТРУДНИК) WHERE(Dept=42). Узнаем максимальную зарплату сотрудников отдела с номером 42).
* Функция SQL AVG - возвращает среднее значение среди всех значений столбца. (**Пример**: SELECT AVG(Years) FROM(СОТРУДНИК) WHERE(Dept=42). Можем узнать средний трудовой стаж сотрудников отдела с номером 42).
* Функция SQL COUNT - возвращает количество записей таблицы базы данных. (Если в запросе указать SELECT COUNT(ИМЯ\_СТОЛБЦА) ..., то результатом будет количество записей без учёта тех записей, в которых значением столбца является NULL (неопределённое). Если использовать в качестве аргумента звёздочку и начать запрос SELECT COUNT(\*) ..., то результатом будет количество всех записей (строк) таблицы.). (**Пример**: SELECT COUNT(Надбавки) FROM(СОТРУДНИК) можем узнать число всех сотрудников, которые получают надбавки к зп (вернет Число сотрудников, у которых значения столбца Надбавки - не NULL), а с запросом SELECT COUNT(\*)(Надбавки) FROM(СОТРУДНИК) можем узнать общее количество записей в таблице.)

1. Агрегатные функции используются с оператором SQL GROUP BY, то есть с группировкой по полям (столбцам) для получения результирующих значений в каждой группе.

Оператор SQL GROUP BY служит для группировки результирующих значений по столбцам таблицы базы данных. (**Пример**: можем найти сумму продаж альбомов (Sale) всех исполнителей (Singer): SELECT Singer, SUM(Sale) AS AllSales FROM Artists GROUP BY Singer. Оператор SQL AS позволяет задать новое имя столбца на выходе.)