Сухачева Валерия Артемовна, 2 курс ИВТ, 1 подгруппа

Инвариантное задание №3.1 по дисциплине «Базы данных»

*ИГРА*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Название** | **Производитель** | **Мин игроков** | **Макс игроков** | **Мин возраст** |
| Диксит | Libellud | 3 | 6 | 8 |
| Диксит | Asmodee | 3 | 6 | 12 |
| Монополия | Hasbro Inc | 2 | 6 |  |
| Jungle | Asmodee | 2 | 4 | 9 |

*КЛИЕНТ*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Ид\_клиента**  **(АК)** | **Фамилия** | **Имя** | **Дата рождения** | **Телефон** | **Почта**  **(АК)** | **Адрес** |
| K1 | Петров | Петр |  | 8-962-587-33-01 | arpet@mail.ru |  |
| K2 | Швыркин | Василий | 10.09.1988 |  | vasvas@gmail.com | ул. Пионерская, 26-58 |

*ЗАКАЗ*

|  |  |  |
| --- | --- | --- |
| **Номер**  **(РК)** | **Получение** | **Создан**  **(АК)** |
| 1 | Самовывоз | 12.09.2017 16:12 |
| 2 | Самовывоз | 12.09.2017 16:14 |
| 3 | Доставка | 13.09.2017 10:09 |

*СОТРУДНИК*

|  |  |  |
| --- | --- | --- |
| **Таб. Номер**  **(АК)** | **Фамилия** | **Имя** |
| С01 | Жуков | Дмитрий |
| С02 | Васичкин | Сергей |

**1. Определите потенциальные ключи для каждого из отношений. Если для каких-то отношений таких ключей окажется несколько, выберите один из них на роль первичного (PK), а остальные пометьте альтернативными (AK).**

РК – Номер заказа

АК – Ид\_клиента, Телефон, Почта, Создан, Таб. Номер

**2. Определите, какие из атрибутов отношений являются обязательными, т.е. не допускают наличие неизвестных значений. Пометьте такие атрибуты как NOT NULL.**

Все, кроме даты рождения(к), телефон(к), адрес(к) и мин.возраста(и)

**3. Определите условия проверки значений для атрибутов (где возможно и, на ваш взгляд, имеет смысл). Условия требуется записать в виде логического выражения (предиката), в левой части которого указано имя атрибута, а в правой – значения, с которыми требуется произвести сравнение. Допускается использование реляционных (>, <, >=, <=, =, !=) и логических (AND, OR) операторов, а также оператора LIKE в том синтаксисе, с которым вы познакомились при выполнении лабораторной работы второй недели**.

SELECT \* FROM ' *ИГРА*' WHERE **Мин игроков** >=3 OR **Макс игроков** <=6 AND **Мин возраст** >=8;

SELECT \* FROM ' *ИГРА*' WHERE **Мин игроков** >=3 OR **Макс игроков** <=6 AND **Мин возраст** >=12;

SELECT \* FROM ' *ИГРА*' WHERE **Мин игроков** >=2 OR **Макс игроков** <=6;

SELECT \* FROM ' *ИГРА*' WHERE **Мин игроков** >=2 OR **Макс игроков** <=4 AND **Мин возраст** >=9;

**4. Определите связи между отношениями.**

ЗАКАЗ, СОТРУДНИК:

* один заказ должен быть обработан ОДНИМ сотрудником
* один сотрудник может вести МНОГО заказов

ЗАКАЗ, КЛИЕНТ:

* один клиент может заказать МНОГО заказов
* каждый номер принадлежит ОДНОМУ клиенту

ЗАКАЗ, ИГРА:

* одна игра может иметь МНОГО заказов
* каждый номер принадлежит ОДНОЙ игре

**5. Установите связь между отношениями, добавив атрибуты первичного ключа главного отношения (того, к которому относится слово ОДИН) в состав дочернего (того, к которому относится слово МНОГО) и отметив их как внешний ключ (FK).**

*При получении связи типа многие-ко-многим, не забудьте, что она должна быть разбита с образованием нового, промежуточного отношения, в которое могут быть добавлены атрибуты, дополнительно характеризующие связь.*

ЗАКАЗ, СОТРУДНИК:

* один заказ должен быть обработан ОДНИМ сотрудником (один к одному)
* один сотрудник может вести МНОГО заказов (один ко многим)

ЗАКАЗ, КЛИЕНТ:

* один клиент может заказать МНОГО заказов (один к многим)
* каждый номер принадлежит ОДНОМУ клиенту (один к одному)

ЗАКАЗ, ИГРА:

* одна игра может иметь МНОГО заказов (один к многим)
* каждый номер принадлежит ОДНОЙ игре (один к одному)

**6. Проверьте добавленные при установке связей атрибуты на обязательность, пометьте их как NOT NULL, если это требуется. Проверьте, не появились ли новые потенциальные ключи в результате добавления новых атрибутов.**

NOT NULL - Мин возраст Дата рождения Телефон (АК) Адрес

**7. Ответьте на вопрос (письменно):**

**Допустим, что каждый сотрудник может обработать только один заказ (после чего его увольняют ). В этом случае, связь между отношениями ЗАКАЗ и СОТРУДНИК трансформируется в один-к-одному.** *Что нужно будет сделать дополнительно для поддержки этой связи? Что может помешать указать одного и того же сотрудника в отношении ЗАКАЗ, ответственным за обработку нескольких разных заказов?*

• Для одного сотрудника будут храниться сведения только по одному заказу

• Один заказ будет идентифицировать только одного сотрудника

Вариационная часть

Задание 3.1: Проанализировать информацию по избыточности баз данных и привести примеры неверного логического проектирования. Предложить алгоритм решения (по устранению) недостатков логической структуры.

Из-за избыточности информации в базе данных возникают не только проблемы модификации, добавления и удаления данных из базы данных, но и остро встает вопрос экономии места на диске, согласитесь глупо хранить одну и ту же информацию в разных местах. Избыточность баз данных тесно связана с нормальными формами. Точнее, информационная избыточность – это отрицательный фактор, влияющий на целостность базы данных, вынуждающий нас приводить свои базы данных к нормальной форме.

Избыточность провоцирует "аномальности обновлений" данных, в результате которых формально правильно составленные запросы к БД смогут выдавать неверные данные. К сказанному есть два важных замечания. Во-первых, избыточность тут подразумевается применительно к логическому описанию данных, в то время как избыточность физического хранения может быть оправданна и разумна. Во-вторых, устранение избыточности, будучи необходимым для "правильного" построения БД, само по себе не гарантирует правильности моделирования предметной области.

Пусть в отношении, представляющем сведения о сотрудниках, есть атрибуты "зарплата", "комиссионные" и "доход". Если по правилам моделируемой предметной области доход сотрудника складывается исключительно из его зарплаты и комиссионных, один из перечисленных атрибутов следует из определения отношения убрать — скорее всего, это будет "доход":

