Тестовое задание

Это тестовое задание не имеет никаких технических ограничений. Ты можешь использовать любые инструменты.

1. Билеты на событие

После успешной покупки билетов на событие, данные попадают в список заказов. Список заказов сохраняется в таблице phpMyAdmin в виде:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** |
| 1 | 003 | 2021-08-21 13:00 | 700 | 1 | 450 | 0 | 11111111 | 00451 | 700 | 2021-01-11 13:22:09 |
| 2 | 006 | 2021-07-29 18:00 | 1000 | 0 | 800 | 2 | 22222222 | 00364 | 1600 | 2021-01-12 16:62:08 |
| 3 | 003 | 2021-08-15 17:00 | 700 | 4 | 450 | 3 | 33333333 | 00015 | 4150 | 2021-01-13 10:08:45 |

где:

id (1) – инкрементальный порядковый номер заказа

event\_id (2) – уникальный id события (у каждого события есть свое название, описание, расписание, цены и свой уникальный event\_id, соответственно)

event\_date (3) – дата и время, на которое были куплены билеты

ticket\_adult\_price (4) – цена взрослого билета на момент покупки

ticket\_adult\_quantity (5) – количество купленных взрослых билетов в этом заказе

ticket\_kid\_price (6) – цена детского билета на момент покупки

ticket\_kid\_quantity (7) – количество купленных детских билетов в этом заказе

barcode (8) – уникальный штрих код заказа

user\_id (9) – уникальный id пользователя

equal\_price (10) – общая сумма заказа

created (11) – дата создания заказа

Вопросы:

1) Некоторые события нужно продавать с дополнительными типами билетов – льготный и групповой, у которых будут свои цены и название. Имеется информация, что вероятно, будут другие типы билетов, которые нужно будет добавить. Нужно уметь сохранять при заказе 2 дополнительных типа билета, льготный и групповой в БД. Задача – показать конечный вид таблицы с добавленными типами билетов. Объяснить свое решение.

2) Часто посетители из одного заказа приходят не одновременно на события. Возникает необходимость чекинить их по отдельности. Для этого у каждого билета должен быть свой баркод. Если в одном заказе было куплено несколько билетов, 2 взрослых, 3 детских, 4 льготных - то должно быть 9 баркодов для каждого билета соответственно. Задача – показать конечный вид таблицы, где у каждого билета свой баркод. Объяснить свое решение.

Ответ:

Я бы поступил следующим образом:

1) Удалил бы из таблицы orders поля \_price и \_quantity

2) Создал бы таблицу categories для разделения билетов на категории, но для простоты предложу использовать ENUM. Предпочтительнее использовать отдельную таблицу, т.к. категории будут более, чем в одной таблице

3) Вынес бы цены из таблицы events в отдельную таблицу prices:

- во-первых для возможности обновления цен с сохранением старых цен (история необходима для аналитики),

- во-вторых для того, чтобы не нарушать 3 нормальную форму при хранении цен для разных категорий

4) Создал бы отдельную таблицу tickets, чтобы при создании заказа, каждый билет создавался отдельно с уникальным баркодом

По итогу в таблице заказов будут хранится только сами заказы. Количество билетов и стоимость заказа можно получить через запрос. В итоге будет соответствие 3 нормальной форме и не будет хранится лишняя информация о тех билетах, которые не приобретались в конкретном заказе. У заказа будет свой баркод, а баркод билетов в заказе будет содержать его часть.

pk – primary key (несколько pk в таблице – составной первичный ключ)

fk – foreign key

Таблица **orders**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **pk** | **fk** |  | **fk** | **fk** |  |
| **id** | **event\_id** | **event\_date** | **barcode** | **user\_id** | **created** |
| 1 | 003 | 2021-08-21 13:00:00 | 11111111 | 00451 | 2021-01-11 13:22:09 |
| 2 | 006 | 2021-07-29 18:00:00 | 22222222 | 00364 | 2021-01-12 16:62:08 |
| 3 | 003 | 2021-08-15 17:00:00 | 33333333 | 00015 | 2021-01-13 10:08:45 |

где:

id – инкрементальный порядковый номер заказа

event\_id – уникальный id события

event\_date – дата и время, на которое были куплены билеты

barcode – уникальный штрих код заказа

user\_id – уникальный id пользователя

created – дата создания заказа

Таблица **prices**

|  |  |  |  |
| --- | --- | --- | --- |
| **Pk, fk** | **pk** | **pk** |  |
| **event\_id** | **category** | **updated\_at** | **price** |
| 003 | adult | 2021-01-11 08:05:33 | 700 |
| 003 | kid | 2021-01-11 08:06:54 | 450 |
| 003 | group | 2021-01-11 08:07:45 | 500 |
| 003 | preferential | 2021-01-11 08:09:09 | 500 |
| 006 | adult | 2021-01-11 08:12:34 | 1000 |
| 006 | kid | 2021-01-11 08:14:21 | 800 |
| 006 | group | 2021-01-11 08:16:01 | 700 |
| 006 | preferential | 2021-01-11 08:17:38 | 600 |

где:

event\_id – уникальный id события

category – категория посетителя (тип данных ENUM = [ ’adult’, ’kid’, ’group’, ’preferential’]), если в перспективе планируется множество типов билетов, можно создать отдельную таблицу categories и заменить это поле на category\_id

updated\_at – дата изменения цены (для сохранения всех изменений цены, поэтому входит в состав ключа)

price – цена

Таблица **tickets**

|  |  |  |  |
| --- | --- | --- | --- |
| **pk, fk** | **pk** |  |  |
| **barcode** | **postfix\_barcode** | **category** | **price** |
| 11111111 | 0001 | adult | 700 |
| 22222222 | 0001 | kid | 800 |
| 22222222 | 0002 | kid | 800 |
| 33333333 | 0001 | adult | 700 |
| 33333333 | 0002 | adult | 700 |
| 33333333 | 0003 | adult | 700 |
| 33333333 | 0004 | adult | 700 |
| 33333333 | 0005 | kid | 450 |
| 33333333 | 0006 | kid | 450 |
| 33333333 | 0007 | kid | 450 |

где:

barcode – уникальный штрих код заказа

postfix\_barcode – постфикс к штрих коду заказа (barcode + postfix\_barcode = штрих код билета)

category – категория посетителя (если тип данных ENUM необходимо контролировать целостность, или создать отдельную таблицу как предлагалось выше)

price – цена на момент покупки (подставляется цена на момент создания заказа согласно категории)