# Отчет о базе данных: Магазин игр

Обзор

Этот отчет объясняет структуру и функциональность базы данных `game\_store`, разработанной для управления данными онлайн-магазина игр. База данных включает таблицы для пользователей, игр, категорий, заказов и других связанных данных. Ниже каждая секция содержит детали о таблицах, примеры данных и выполненные запросы.

## Описание таблиц

### Пользователь

Назначение: Хранит информацию о пользователях (покупателях).

Поля:

ID: Уникальный идентификатор каждого пользователя.

name: Имя пользователя.

email: Электронная почта пользователя.

Пример данных:

INSERT INTO User (name, email) VALUES ('Alex', 'alex@example.com');

### Адрес

Назначение: Хранит информацию об адресах пользователей.

Поля:

ID: Уникальный идентификатор адреса.

user\_id: Связывает адрес с конкретным пользователем.

city, street, house\_number, postal\_code: Различные детали адреса.

Пример данных:

INSERT INTO Address (user\_id, city, street, house\_number, postal\_code) VALUES (1, 'Москва', 'Ленина', '10', '101000');

### Метод\_Оплаты

Назначение: Хранит доступные методы оплаты.

Поля:

ID: Уникальный идентификатор метода оплаты.

method: Описание метода оплаты (например, Кредитная карта, PayPal).

Пример данных:

INSERT INTO Payment\_Method (method) VALUES ('Кредитная карта');

### Категория

Назначение: Классифицирует игры по категориям.

Поля:

ID: Уникальный идентификатор категории.

name: Название категории (например, Экшн, РПГ).

Пример данных:

INSERT INTO Category (name) VALUES ('Экшн');

### Игра

Назначение: Хранит информацию об играх.

Поля:

ID: Уникальный идентификатор игры.

title, developer, price: Детали игры.

category\_id: Идентификатор, связывающий игру с категорией.

Пример данных:

INSERT INTO Game (title, developer, price, category\_id) VALUES ('Игра A', 'Разработчик A', 199.99, 1);

### Скидка

Назначение: Хранит информацию о скидках.

Поля:

ID: Уникальный идентификатор скидки.

description: Описание скидки.

percentage: Процент скидки.

Пример данных:

INSERT INTO Discount (description, percentage) VALUES ('Праздничная скидка', 10);

### Заказ

Назначение: Хранит информацию о заказах пользователей.

Поля:

ID: Уникальный идентификатор заказа.

user\_id: Идентификатор пользователя, оформившего заказ.

order\_date: Дата заказа.

total\_amount: Сумма заказа.

payment\_method\_id: Идентификатор метода оплаты.

Пример данных:

INSERT INTO Order (user\_id, order\_date, total\_amount, payment\_method\_id) VALUES (1, '2024-10-01', 259.98, 1);

### Детали\_Заказа

Назначение: Хранит детали каждой позиции заказа.

Поля:

ID: Уникальный идентификатор позиции заказа.

order\_id: Идентификатор заказа.

game\_id: Идентификатор игры в заказе.

quantity: Количество игры в заказе.

price: Цена игры в заказе.

Пример данных:

INSERT INTO Order\_Details (order\_id, game\_id, quantity, price) VALUES (1, 1, 1, 199.99);

### Скидка\_На\_Заказ

Назначение: Связывает заказы с применяемыми скидками.

Поля:

ID: Уникальный идентификатор связи скидки с заказом.

order\_id: Идентификатор заказа.

discount\_id: Идентификатор примененной скидки.

Пример данных:

INSERT INTO Order\_Discount (order\_id, discount\_id) VALUES (1, 1);

### Отзыв

Назначение: Хранит отзывы пользователей об играх.

Поля:

ID: Уникальный идентификатор отзыва.

user\_id: Идентификатор пользователя, оставившего отзыв.

game\_id: Идентификатор игры, на которую оставлен отзыв.

rating: Оценка (от 1 до 5).

comment: Текст комментария пользователя.

Пример данных:

INSERT INTO Review (user\_id, game\_id, rating, comment) VALUES (1, 1, 5, 'Отличная игра!');

## Примеры запросов

### Фильтрация и Сортировка

Описание: Выводит заказы, сделанные в 2024 году, отсортированные по дате в порядке убывания.

SQL Запрос:

SELECT \* FROM Order WHERE order\_date BETWEEN '2024-01-01' AND '2024-12-31' ORDER BY order\_date DESC;

### Группировка и Фильтрация по Агрегату (HAVING)

Описание: Группирует заказы по пользователям и отображает пользователей с более чем одним заказом.

SQL Запрос:

SELECT user\_id, COUNT(\*) AS order\_count FROM Order GROUP BY user\_id HAVING COUNT(\*) > 1;

### Уникальные значения и Сопоставление шаблона

Описание: Выводит уникальные названия игр, начинающиеся с 'Game'.

SQL Запрос:

SELECT DISTINCT title FROM Game WHERE title LIKE 'Game%';

## Агрегатные функции

Средняя цена игры: SELECT AVG(price) AS average\_price FROM Game;

Максимальная сумма заказа: SELECT MAX(total\_amount) AS max\_total FROM Order;

Общее количество заказов: SELECT COUNT(\*) AS total\_orders FROM Order;

## Пример Оконной функции

Описание: Выводит все заказы и добавляет столбец со средним значением заказа с использованием оконной функции.

SQL Запрос:

SELECT ID, user\_id, order\_date, total\_amount, AVG(total\_amount) OVER () AS average\_order\_total FROM Order;