# Структура базы данных для ИС точки проката "Игора"

## Основные таблицы (сущности)

### 1. Users (Пользователи системы)

**Описание:** Сотрудники точки проката

Users

├── user\_id (PK) - INT, AUTO\_INCREMENT

├── login - VARCHAR(50), UNIQUE, NOT NULL

├── password\_hash - VARCHAR(255), NOT NULL

├── first\_name - VARCHAR(100), NOT NULL

├── last\_name - VARCHAR(100), NOT NULL

├── middle\_name - VARCHAR(100)

├── role\_id (FK) - INT, NOT NULL

├── photo\_path - VARCHAR(500)

├── is\_active - BOOLEAN, DEFAULT TRUE

├── created\_at - TIMESTAMP, DEFAULT CURRENT\_TIMESTAMP

└── updated\_at - TIMESTAMP, DEFAULT CURRENT\_TIMESTAMP ON UPDATE

### 2. Roles (Роли пользователей)

**Описание:** Роли в системе (продавец, старший смены, администратор)

Roles

├── role\_id (PK) - INT, AUTO\_INCREMENT

├── role\_name - VARCHAR(50), UNIQUE, NOT NULL

├── role\_description - TEXT

└── permissions - JSON

### 3. Clients (Клиенты)

**Описание:** Клиенты точки проката

Clients

├── client\_id (PK) - INT, AUTO\_INCREMENT

├── client\_code - VARCHAR(20), UNIQUE, NOT NULL

├── first\_name - VARCHAR(100), NOT NULL

├── last\_name - VARCHAR(100), NOT NULL

├── middle\_name - VARCHAR(100)

├── email - VARCHAR(255), UNIQUE

├── phone - VARCHAR(20)

├── address - TEXT

├── birth\_date - DATE

├── passport\_series - VARCHAR(10)

├── passport\_number - VARCHAR(20)

├── created\_at - TIMESTAMP, DEFAULT CURRENT\_TIMESTAMP

└── updated\_at - TIMESTAMP, DEFAULT CURRENT\_TIMESTAMP ON UPDATE

### 4. Equipment\_Categories (Категории оборудования)

**Описание:** Категории проката (лыжи, сноуборды, ботинки и т.д.)

Equipment\_Categories

├── category\_id (PK) - INT, AUTO\_INCREMENT

├── category\_name - VARCHAR(100), NOT NULL

├── category\_description - TEXT

└── is\_active - BOOLEAN, DEFAULT TRUE

### 5. Equipment (Оборудование)

**Описание:** Конкретные единицы оборудования

Equipment

├── equipment\_id (PK) - INT, AUTO\_INCREMENT

├── category\_id (FK) - INT, NOT NULL

├── brand - VARCHAR(100)

├── model - VARCHAR(100)

├── size - VARCHAR(20)

├── condition\_status - ENUM('excellent', 'good', 'satisfactory', 'needs\_repair')

├── purchase\_date - DATE

├── last\_maintenance\_date - DATE

├── is\_available - BOOLEAN, DEFAULT TRUE

├── barcode - VARCHAR(255), UNIQUE

└── notes - TEXT

### 6. Services (Услуги проката)

**Описание:** Услуги, предоставляемые точкой проката

Services

├── service\_id (PK) - INT, AUTO\_INCREMENT

├── service\_name - VARCHAR(200), NOT NULL

├── service\_description - TEXT

├── category\_id (FK) - INT

├── hourly\_rate - DECIMAL(10,2), NOT NULL

├── daily\_rate - DECIMAL(10,2)

├── deposit\_amount - DECIMAL(10,2)

├── is\_active - BOOLEAN, DEFAULT TRUE

├── created\_at - TIMESTAMP, DEFAULT CURRENT\_TIMESTAMP

└── updated\_at - TIMESTAMP, DEFAULT CURRENT\_TIMESTAMP ON UPDATE

### 7. Orders (Заказы)

**Описание:** Основная таблица заказов

Orders

├── order\_id (PK) - INT, AUTO\_INCREMENT

├── order\_number - VARCHAR(50), UNIQUE, NOT NULL

├── client\_id (FK) - INT, NOT NULL

├── user\_id (FK) - INT, NOT NULL (кто создал заказ)

├── order\_date - TIMESTAMP, DEFAULT CURRENT\_TIMESTAMP

├── start\_date - DATETIME, NOT NULL

├── end\_date - DATETIME, NOT NULL

├── total\_amount - DECIMAL(12,2), NOT NULL

├── deposit\_amount - DECIMAL(10,2)

├── status - ENUM('active', 'completed', 'cancelled', 'archived')

├── barcode - VARCHAR(255), UNIQUE

├── notes - TEXT

└── created\_at - TIMESTAMP, DEFAULT CURRENT\_TIMESTAMP

### 8. Order\_Services (Услуги в заказе)

**Описание:** Связь заказов с услугами (многие ко многим)

Order\_Services

├── order\_service\_id (PK) - INT, AUTO\_INCREMENT

├── order\_id (FK) - INT, NOT NULL

├── service\_id (FK) - INT, NOT NULL

├── equipment\_id (FK) - INT

├── quantity - INT, NOT NULL, DEFAULT 1

├── unit\_price - DECIMAL(10,2), NOT NULL

├── total\_price - DECIMAL(10,2), NOT NULL

├── rental\_hours - INT

└── notes - TEXT

### 9. Equipment\_Returns (Возврат оборудования)

**Описание:** Информация о возврате оборудования

Equipment\_Returns

├── return\_id (PK) - INT, AUTO\_INCREMENT

├── order\_id (FK) - INT, NOT NULL

├── equipment\_id (FK) - INT, NOT NULL

├── returned\_by\_user\_id (FK) - INT, NOT NULL

├── return\_date - TIMESTAMP, DEFAULT CURRENT\_TIMESTAMP

├── condition\_on\_return - ENUM('excellent', 'good', 'satisfactory', 'damaged')

├── damage\_description - TEXT

├── additional\_charges - DECIMAL(10,2), DEFAULT 0

└── notes - TEXT

### 10. Login\_History (История входов)

**Описание:** Журнал входов в систему

Login\_History

├── history\_id (PK) - INT, AUTO\_INCREMENT

├── user\_login - VARCHAR(50), NOT NULL

├── attempt\_time - TIMESTAMP, DEFAULT CURRENT\_TIMESTAMP

├── is\_successful - BOOLEAN, NOT NULL

├── ip\_address - VARCHAR(45)

├── user\_agent - TEXT

└── failure\_reason - VARCHAR(255)

### 11. Session\_Management (Управление сеансами)

**Описание:** Активные сеансы пользователей

Session\_Management

├── session\_id (PK) - VARCHAR(255)

├── user\_id (FK) - INT, NOT NULL

├── login\_time - TIMESTAMP, DEFAULT CURRENT\_TIMESTAMP

├── last\_activity - TIMESTAMP, DEFAULT CURRENT\_TIMESTAMP ON UPDATE

├── session\_duration\_minutes - INT, DEFAULT 150

├── is\_active - BOOLEAN, DEFAULT TRUE

└── logout\_time - TIMESTAMP

### 12. Consumables (Расходные материалы)

**Описание:** Расходные материалы точки проката

Consumables

├── consumable\_id (PK) - INT, AUTO\_INCREMENT

├── item\_name - VARCHAR(200), NOT NULL

├── item\_description - TEXT

├── unit\_of\_measure - VARCHAR(20)

├── current\_stock - DECIMAL(10,2), NOT NULL, DEFAULT 0

├── minimum\_stock - DECIMAL(10,2), DEFAULT 0

├── unit\_cost - DECIMAL(10,2)

├── supplier - VARCHAR(200)

├── last\_updated - TIMESTAMP, DEFAULT CURRENT\_TIMESTAMP ON UPDATE

└── is\_active - BOOLEAN, DEFAULT TRUE

### 13. Consumable\_Transactions (Движение расходных материалов)

**Описание:** Приход и расход материалов

Consumable\_Transactions

├── transaction\_id (PK) - INT, AUTO\_INCREMENT

├── consumable\_id (FK) - INT, NOT NULL

├── user\_id (FK) - INT, NOT NULL

├── transaction\_type - ENUM('receipt', 'consumption', 'writeoff')

├── quantity - DECIMAL(10,2), NOT NULL

├── transaction\_date - TIMESTAMP, DEFAULT CURRENT\_TIMESTAMP

├── reason - TEXT

├── document\_number - VARCHAR(100)

└── notes - TEXT

### 14. Reports\_Cache (Кэш отчетов)

**Описание:** Временное хранение сгенерированных отчетов

Reports\_Cache

├── cache\_id (PK) - INT, AUTO\_INCREMENT

├── report\_type - VARCHAR(100), NOT NULL

├── parameters\_hash - VARCHAR(255), NOT NULL

├── report\_data - LONGTEXT

├── file\_path - VARCHAR(500)

├── generated\_by\_user\_id (FK) - INT, NOT NULL

├── generated\_at - TIMESTAMP, DEFAULT CURRENT\_TIMESTAMP

└── expires\_at - TIMESTAMP

## Связи между таблицами (Foreign Keys)

### Основные связи:

#### 1. Users → Roles (Многие к одному)

ALTER TABLE Users

ADD CONSTRAINT FK\_Users\_Roles

FOREIGN KEY (role\_id) REFERENCES Roles(role\_id);

#### 2. Equipment → Equipment\_Categories (Многие к одному)

ALTER TABLE Equipment

ADD CONSTRAINT FK\_Equipment\_Categories

FOREIGN KEY (category\_id) REFERENCES Equipment\_Categories(category\_id);

#### 3. Services → Equipment\_Categories (Многие к одному)

ALTER TABLE Services

ADD CONSTRAINT FK\_Services\_Categories

FOREIGN KEY (category\_id) REFERENCES Equipment\_Categories(category\_id);

#### 4. Orders → Clients (Многие к одному)

ALTER TABLE Orders

ADD CONSTRAINT FK\_Orders\_Clients

FOREIGN KEY (client\_id) REFERENCES Clients(client\_id);

#### 5. Orders → Users (Многие к одному)

ALTER TABLE Orders

ADD CONSTRAINT FK\_Orders\_Users

FOREIGN KEY (user\_id) REFERENCES Users(user\_id);

#### 6. Order\_Services → Orders (Многие к одному)

ALTER TABLE Order\_Services

ADD CONSTRAINT FK\_OrderServices\_Orders

FOREIGN KEY (order\_id) REFERENCES Orders(order\_id) ON DELETE CASCADE;

#### 7. Order\_Services → Services (Многие к одному)

ALTER TABLE Order\_Services

ADD CONSTRAINT FK\_OrderServices\_Services

FOREIGN KEY (service\_id) REFERENCES Services(service\_id);

#### 8. Order\_Services → Equipment (Многие к одному)

ALTER TABLE Order\_Services

ADD CONSTRAINT FK\_OrderServices\_Equipment

FOREIGN KEY (equipment\_id) REFERENCES Equipment(equipment\_id);

#### 9. Equipment\_Returns → Orders (Многие к одному)

ALTER TABLE Equipment\_Returns

ADD CONSTRAINT FK\_Returns\_Orders

FOREIGN KEY (order\_id) REFERENCES Orders(order\_id);

#### 10. Equipment\_Returns → Equipment (Многие к одному)

ALTER TABLE Equipment\_Returns

ADD CONSTRAINT FK\_Returns\_Equipment

FOREIGN KEY (equipment\_id) REFERENCES Equipment(equipment\_id);

#### 11. Equipment\_Returns → Users (Многие к одному)

ALTER TABLE Equipment\_Returns

ADD CONSTRAINT FK\_Returns\_Users

FOREIGN KEY (returned\_by\_user\_id) REFERENCES Users(user\_id);

#### 12. Session\_Management → Users (Многие к одному)

ALTER TABLE Session\_Management

ADD CONSTRAINT FK\_Sessions\_Users

FOREIGN KEY (user\_id) REFERENCES Users(user\_id);

#### 13. Consumable\_Transactions → Consumables (Многие к одному)

ALTER TABLE Consumable\_Transactions

ADD CONSTRAINT FK\_Transactions\_Consumables

FOREIGN KEY (consumable\_id) REFERENCES Consumables(consumable\_id);

#### 14. Consumable\_Transactions → Users (Многие к одному)

ALTER TABLE Consumable\_Transactions

ADD CONSTRAINT FK\_Transactions\_Users

FOREIGN KEY (user\_id) REFERENCES Users(user\_id);

#### 15. Reports\_Cache → Users (Многие к одному)

ALTER TABLE Reports\_Cache

ADD CONSTRAINT FK\_Reports\_Users

FOREIGN KEY (generated\_by\_user\_id) REFERENCES Users(user\_id);

## Индексы для оптимизации

### Основные индексы:

-- Поиск клиентов

CREATE INDEX idx\_clients\_email ON Clients(email);

CREATE INDEX idx\_clients\_phone ON Clients(phone);

CREATE INDEX idx\_clients\_name ON Clients(last\_name, first\_name);

CREATE INDEX idx\_clients\_code ON Clients(client\_code);

-- Поиск заказов

CREATE INDEX idx\_orders\_date ON Orders(order\_date);

CREATE INDEX idx\_orders\_status ON Orders(status);

CREATE INDEX idx\_orders\_number ON Orders(order\_number);

CREATE INDEX idx\_orders\_client ON Orders(client\_id, order\_date);

-- История входов

CREATE INDEX idx\_login\_history\_time ON Login\_History(attempt\_time);

CREATE INDEX idx\_login\_history\_user ON Login\_History(user\_login);

CREATE INDEX idx\_login\_history\_success ON Login\_History(is\_successful, attempt\_time);

-- Оборудование

CREATE INDEX idx\_equipment\_available ON Equipment(is\_available);

CREATE INDEX idx\_equipment\_category ON Equipment(category\_id);

CREATE INDEX idx\_equipment\_barcode ON Equipment(barcode);

-- Услуги в заказе

CREATE INDEX idx\_order\_services\_order ON Order\_Services(order\_id);

CREATE INDEX idx\_order\_services\_service ON Order\_Services(service\_id);

-- Возврат оборудования

CREATE INDEX idx\_returns\_date ON Equipment\_Returns(return\_date);

CREATE INDEX idx\_returns\_order ON Equipment\_Returns(order\_id);

-- Сеансы

CREATE INDEX idx\_sessions\_active ON Session\_Management(is\_active, last\_activity);

CREATE INDEX idx\_sessions\_user ON Session\_Management(user\_id);

## Типовые данные для заполнения

### Роли:

INSERT INTO Roles (role\_name, role\_description) VALUES

('Продавец', 'Базовые права: формирование заказов'),

('Старший смены', 'Расширенные права: формирование заказов + прием товара'),

('Администратор', 'Полные права: все функции системы');

### Категории оборудования:

INSERT INTO Equipment\_Categories (category\_name, category\_description) VALUES

('Горные лыжи', 'Лыжи для горнолыжного спорта'),

('Сноуборды', 'Доски для сноубординга'),

('Лыжные ботинки', 'Ботинки для горных лыж'),

('Ботинки для сноуборда', 'Ботинки для сноубординга'),

('Шлемы', 'Защитные шлемы'),

('Защита', 'Защитные элементы (наколенники, налокотники)'),

('Палки', 'Лыжные палки'),

('Ватрушки', 'Надувные санки для катания');

### Примеры услуг:

INSERT INTO Services (service\_name, service\_description, category\_id, hourly\_rate, daily\_rate, deposit\_amount) VALUES

('Прокат горных лыж', 'Аренда комплекта горных лыж', 1, 300.00, 1500.00, 5000.00),

('Прокат сноуборда', 'Аренда сноуборда', 2, 350.00, 1800.00, 5500.00),

('Прокат лыжных ботинок', 'Аренда лыжных ботинок', 3, 150.00, 800.00, 2000.00),

('Прокат ботинок для сноуборда', 'Аренда ботинок для сноуборда', 4, 150.00, 800.00, 2000.00),

('Прокат шлема', 'Аренда защитного шлема', 5, 100.00, 500.00, 1500.00),

('Прокат ватрушки', 'Аренда ватрушки для катания', 8, 200.00, 1000.00, 1000.00);

## Ограничения и проверки

### Check-ограничения:

-- Проверка дат в заказах

ALTER TABLE Orders

ADD CONSTRAINT chk\_orders\_dates

CHECK (end\_date > start\_date);

-- Проверка положительных сумм

ALTER TABLE Orders

ADD CONSTRAINT chk\_orders\_amount

CHECK (total\_amount >= 0);

ALTER TABLE Order\_Services

ADD CONSTRAINT chk\_orderservices\_quantity

CHECK (quantity > 0);

ALTER TABLE Order\_Services

ADD CONSTRAINT chk\_orderservices\_price

CHECK (unit\_price >= 0 AND total\_price >= 0);

-- Проверка остатков материалов

ALTER TABLE Consumables

ADD CONSTRAINT chk\_consumables\_stock

CHECK (current\_stock >= 0 AND minimum\_stock >= 0);

-- Проверка email

ALTER TABLE Clients

ADD CONSTRAINT chk\_clients\_email

CHECK (email REGEXP '^[A-Za-z0-9.\_%+-]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,});

### Триггеры для бизнес-логики:

#### 1. Обновление остатков оборудования:

DELIMITER //

CREATE TRIGGER tr\_update\_equipment\_availability

AFTER INSERT ON Order\_Services

FOR EACH ROW

BEGIN

IF NEW.equipment\_id IS NOT NULL THEN

UPDATE Equipment

SET is\_available = FALSE

WHERE equipment\_id = NEW.equipment\_id;

END IF;

END//

DELIMITER ;

#### 2. Возврат оборудования:

DELIMITER //

CREATE TRIGGER tr\_return\_equipment

AFTER INSERT ON Equipment\_Returns

FOR EACH ROW

BEGIN

UPDATE Equipment

SET is\_available = TRUE,

condition\_status = NEW.condition\_on\_return

WHERE equipment\_id = NEW.equipment\_id;

UPDATE Orders

SET status = 'completed'

WHERE order\_id = NEW.order\_id;

END//

DELIMITER ;

#### 3. Автоматический расчет общей стоимости услуг в заказе:

DELIMITER //

CREATE TRIGGER tr\_calculate\_order\_total

AFTER INSERT ON Order\_Services

FOR EACH ROW

BEGIN

UPDATE Orders

SET total\_amount = (

SELECT COALESCE(SUM(total\_price), 0)

FROM Order\_Services

WHERE order\_id = NEW.order\_id

)

WHERE order\_id = NEW.order\_id;

END//

DELIMITER ;

## Представления (Views) для отчетности

### 1. Активные заказы:

CREATE VIEW v\_active\_orders AS

SELECT

o.order\_id,

o.order\_number,

CONCAT(c.last\_name, ' ', c.first\_name) as client\_name,

c.phone,

o.start\_date,

o.end\_date,

o.total\_amount,

o.status

FROM Orders o

JOIN Clients c ON o.client\_id = c.client\_id

WHERE o.status = 'active';

### 2. Статистика по дням:

CREATE VIEW v\_daily\_statistics AS

SELECT

DATE(order\_date) as order\_day,

COUNT(\*) as orders\_count,

SUM(total\_amount) as daily\_revenue,

COUNT(DISTINCT client\_id) as unique\_clients

FROM Orders

WHERE status != 'cancelled'

GROUP BY DATE(order\_date);

### 3. Популярные услуги:

CREATE VIEW v\_popular\_services AS

SELECT

s.service\_name,

COUNT(os.order\_service\_id) as booking\_count,

SUM(os.total\_price) as total\_revenue,

AVG(os.unit\_price) as avg\_price

FROM Services s

JOIN Order\_Services os ON s.service\_id = os.service\_id

JOIN Orders o ON os.order\_id = o.order\_id

WHERE o.status != 'cancelled'

GROUP BY s.service\_id, s.service\_name

ORDER BY booking\_count DESC;

Эта структура базы данных обеспечивает:

* **3-ю нормальную форму** - избегание избыточности данных
* **Ссылочную целостность** - через внешние ключи
* **Гибкость** - возможность расширения функционала
* **Производительность** - через индексы и оптимизированные запросы
* **Бизнес-логику** - через триггеры и процедуры