

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