# 데이터베이스 설계서

## 📋 문서 개요

### 작성 목적

본 데이터베이스 설계서는 금융 투자 플랫폼의 데이터 구조와 관계를 체계적으로 정의하고, 개발팀과 운영팀이 공통된 이해를 바탕으로 시스템을 구축하고 유지보수할 수 있도록 작성되었습니다.

### 작성 배경

* **시스템 복잡도 증가**: 계정 관리, 포트폴리오, AI 채팅, 자동매매 등 다양한 도메인의 통합 관리 필요
* **확장성 요구**: 사용자 증가에 따른 수평적 확장 가능한 아키텍처 설계 필요
* **성능 최적화**: 대용량 금융 데이터의 효율적 처리를 위한 샤딩 전략 수립
* **보안 강화**: 금융 정보 보호를 위한 체계적인 보안 설계 요구

### 기대 효과

1. **개발 효율성 향상**: 명확한 데이터 구조 정의로 개발 시간 단축
2. **유지보수 용이성**: 체계적인 문서화로 시스템 이해도 향상
3. **확장성 확보**: 샤딩 기반 아키텍처로 무한 확장 가능
4. **데이터 무결성**: 명확한 제약조건과 관계 정의로 데이터 품질 보장
5. **팀 간 커뮤니케이션**: 공통된 데이터 모델 이해로 협업 효율 증대

### 문서 활용 방안

* **개발자**: 테이블 구조, API 흐름, 스토어드 프로시저 참조
* **DBA**: 인덱싱 전략, 파티셔닝, 백업 정책 수립
* **아키텍트**: 시스템 확장 계획, 성능 최적화 방안 도출
* **보안팀**: 데이터 암호화, 접근 제어 정책 검토

## 📑 목차

1. [개요](#1-개요)
2. [ERD (Entity-Relationship Diagram)](#2-erd-entity-relationship-diagram)
   * 2.1 전체 ERD 개요
   * 2.2 시스템 아키텍처 다이어그램
   * 2.3 글로벌 DB ERD
   * 2.4 샤드 DB ERD - 계좌 및 포트폴리오 도메인
   * 2.5 샤드 DB ERD - AI 채팅 및 자동매매 도메인
   * 2.6 샤드 DB ERD - 알림 및 시장 데이터 도메인
   * 2.7 AI 채팅 도메인 상세 ERD
   * 2.8 자동매매 도메인 상세 ERD
3. [주요 테이블 정의서](#3-주요-테이블-정의서)
4. [API 흐름도 및 데이터 접근 패턴](#4-api-흐름도-및-데이터-접근-패턴)
5. [제약조건 명세](#5-제약조건-명세)
6. [쿼리 성능 고려사항](#6-쿼리-성능-고려사항)
7. [데이터 보안 고려사항](#7-데이터-보안-고려사항)
8. [백업 및 복구 전략](#8-백업-및-복구-전략)
9. [모니터링 및 유지보수](#9-모니터링-및-유지보수)
10. [확장성 고려사항](#10-확장성-고려사항)

## 

## 1. 개요

### 1.1 목적

본 문서는 금융 투자 플랫폼 서비스에서 필요한 주요 데이터 구조를 정의하고, 대용량 트래픽 처리를 위한 샤딩 기반의 확장 가능한 데이터베이스 시스템을 구축하기 위한 설계 지침을 제공합니다.

### 1.2 범위

* 샤딩 기반 분산 데이터베이스 아키텍처
* 글로벌 DB와 샤드 DB의 역할 분리
* 스토어드 프로시저 기반 데이터 접근
* 보안 및 개인정보 보호 방안

### 1.3 대상 독자

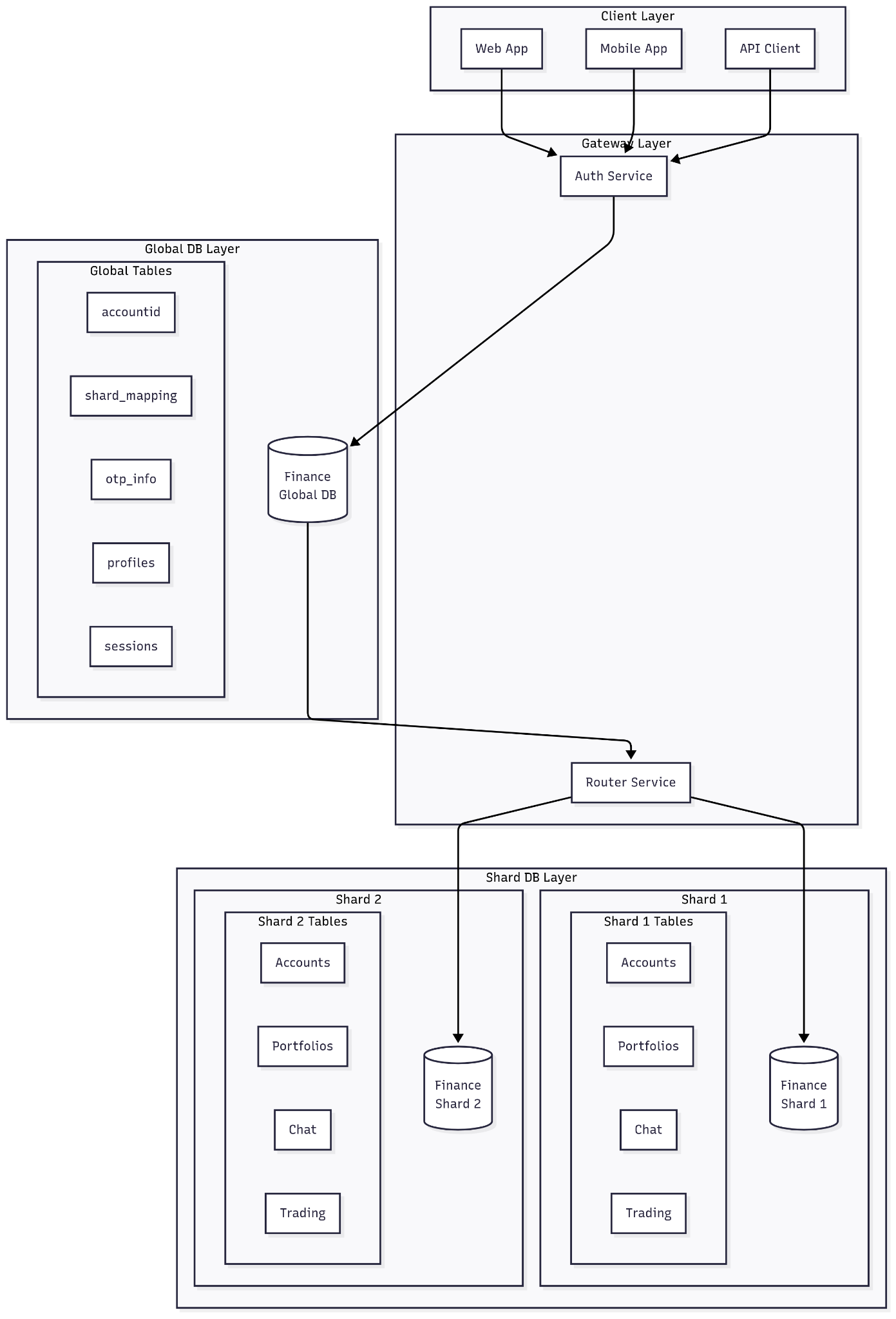
* 개발팀
* 데이터베이스 관리자
* 시스템 아키텍트
* 보안 담당자

## 2. ERD (Entity-Relationship Diagram)

### 2.1 전체 ERD 개요

시스템은 크게 글로벌 DB와 샤드 DB로 구분되며, 사용자 데이터는 account\_db\_key를 기준으로 샤딩됩니다.

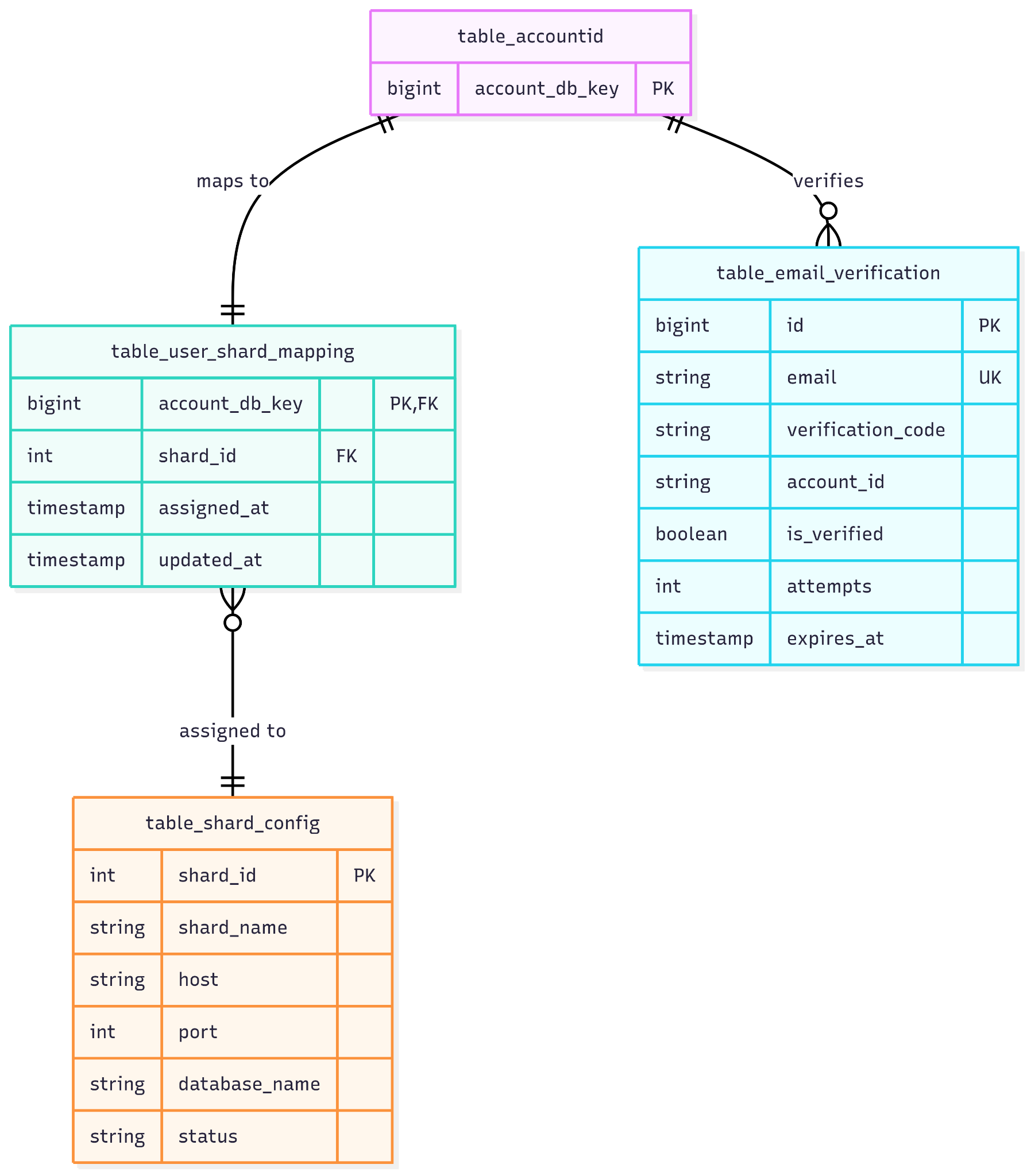
### 2.2 시스템 아키텍처 다이어그램

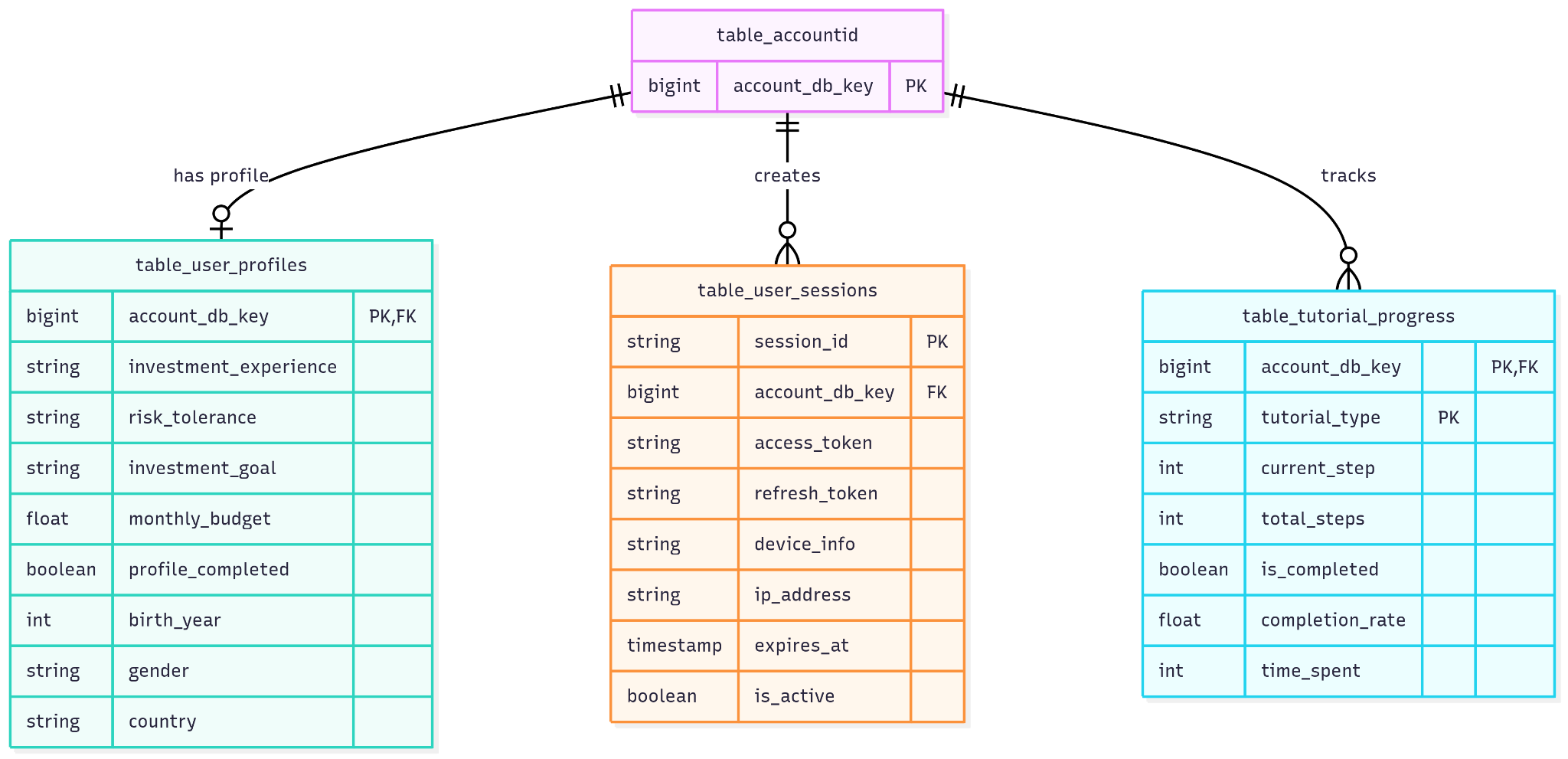
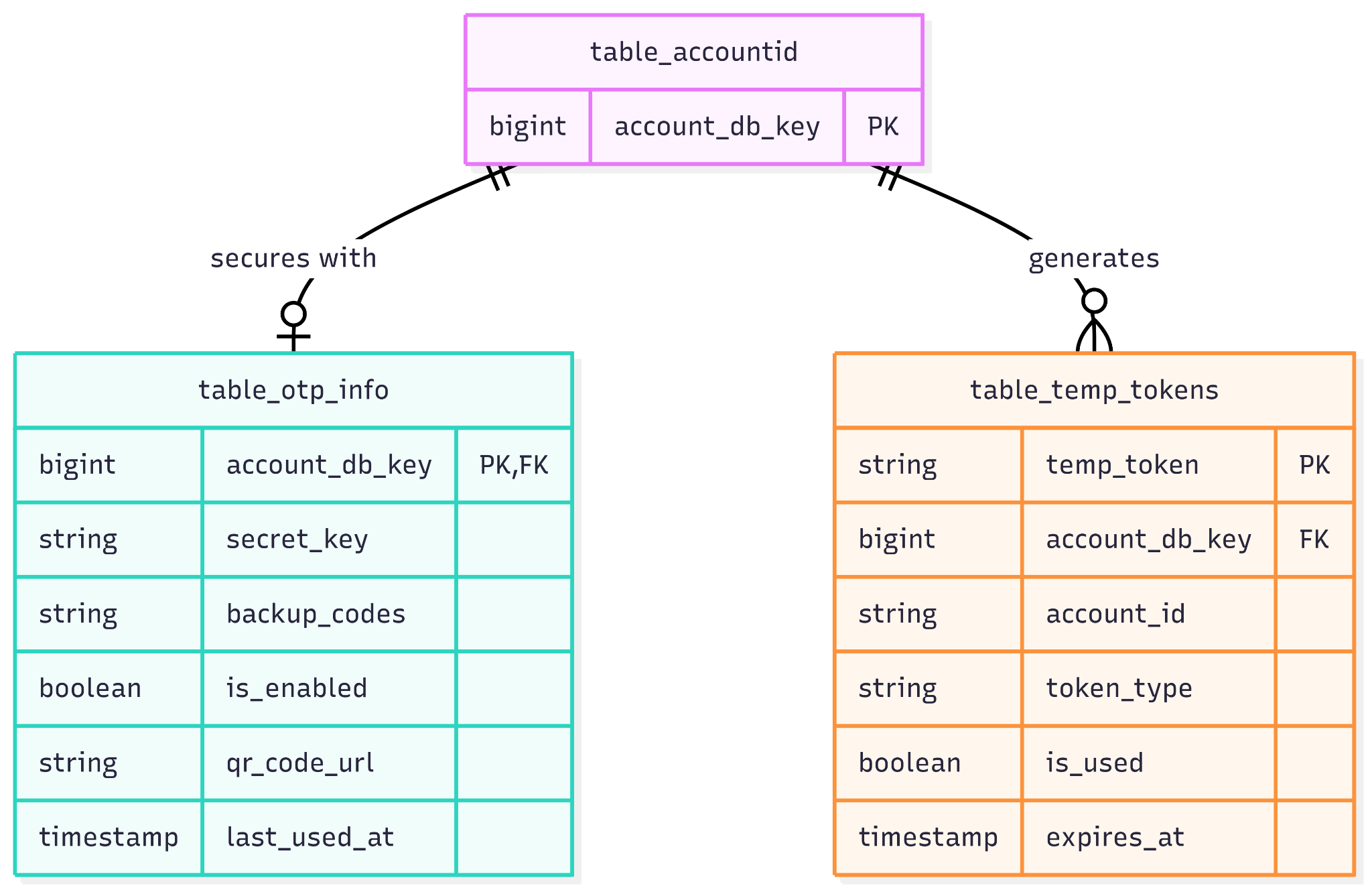
전체 시스템의 데이터 흐름을 보여주는 아키텍처입니다.

**아키텍처 특징:**

* **수평적 확장**: 샤드 추가로 용량 확장 가능
* **독립적 운영**: 샤드별 독립적인 장애 격리
* **중앙 인증**: Global DB에서 통합 인증 관리
* **효율적 라우팅**: account\_db\_key 기반 샤드 자동 라우팅

### 2.3 글로벌 DB ERD (finance\_global)

인증, 계정 관리, 샤드 매핑 등 전역 데이터를 관리하는 글로벌 DB의 ERD입니다. 모든 사용자의 기본 정보와 샤드 위치 정보를 중앙에서 관리합니다.

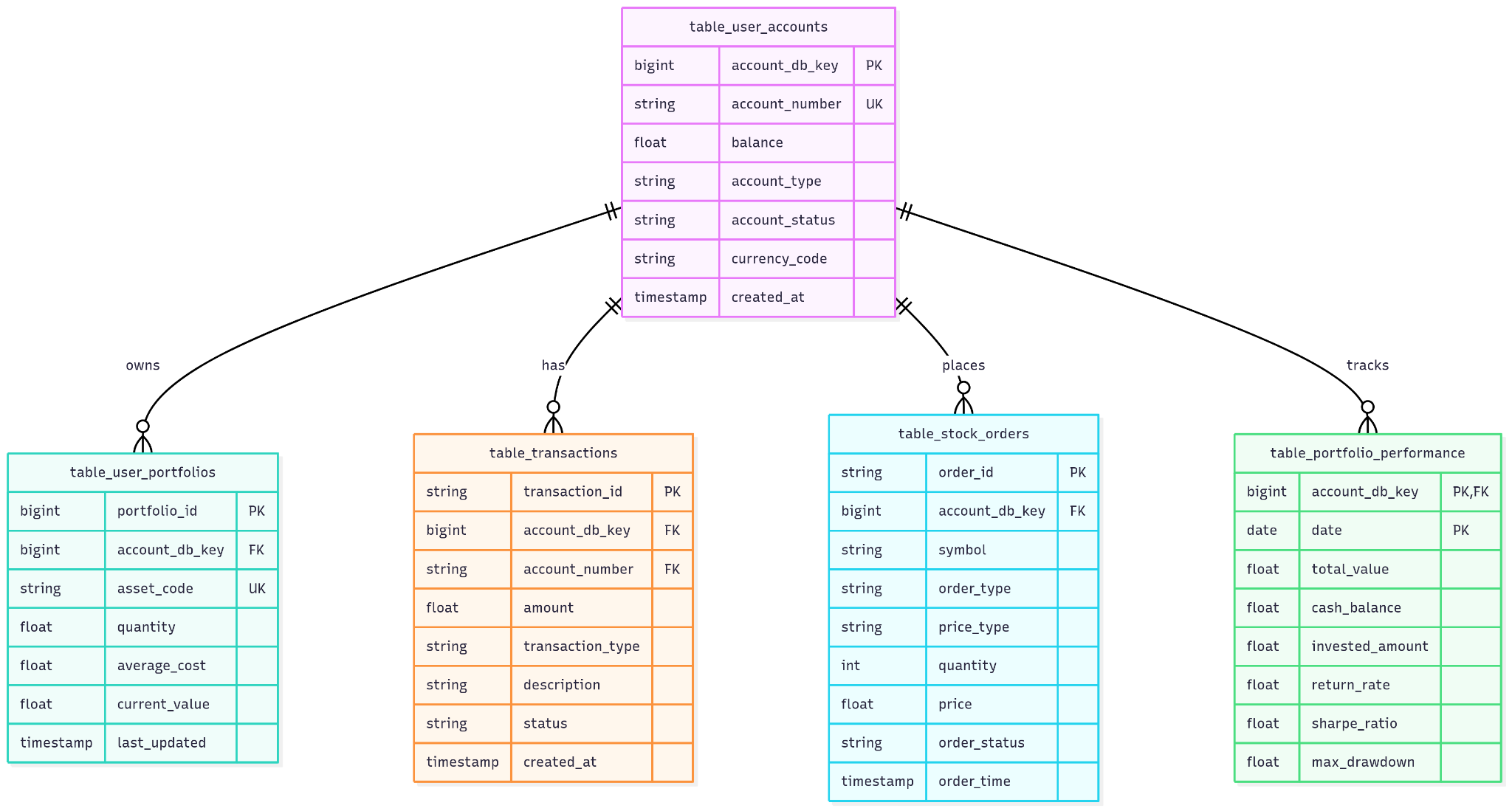


**주요 관계 설명:**

* table\_accountid가 중심 엔터티로 모든 사용자 계정 정보 보유
* table\_user\_shard\_mapping을 통해 사용자별 샤드 위치 관리
* table\_otp\_info, table\_user\_profiles 등은 1:1 관계로 사용자 부가 정보 저장
* table\_user\_sessions, table\_temp\_tokens는 1:N 관계로 다중 세션/토큰 관리

### 2.4 샤드 DB ERD - 계좌 및 포트폴리오 도메인

사용자의 금융 계좌 정보와 포트폴리오를 관리하는 핵심 테이블들입니다. account\_db\_key를 기준으로 샤드별로 분산 저장됩니다.

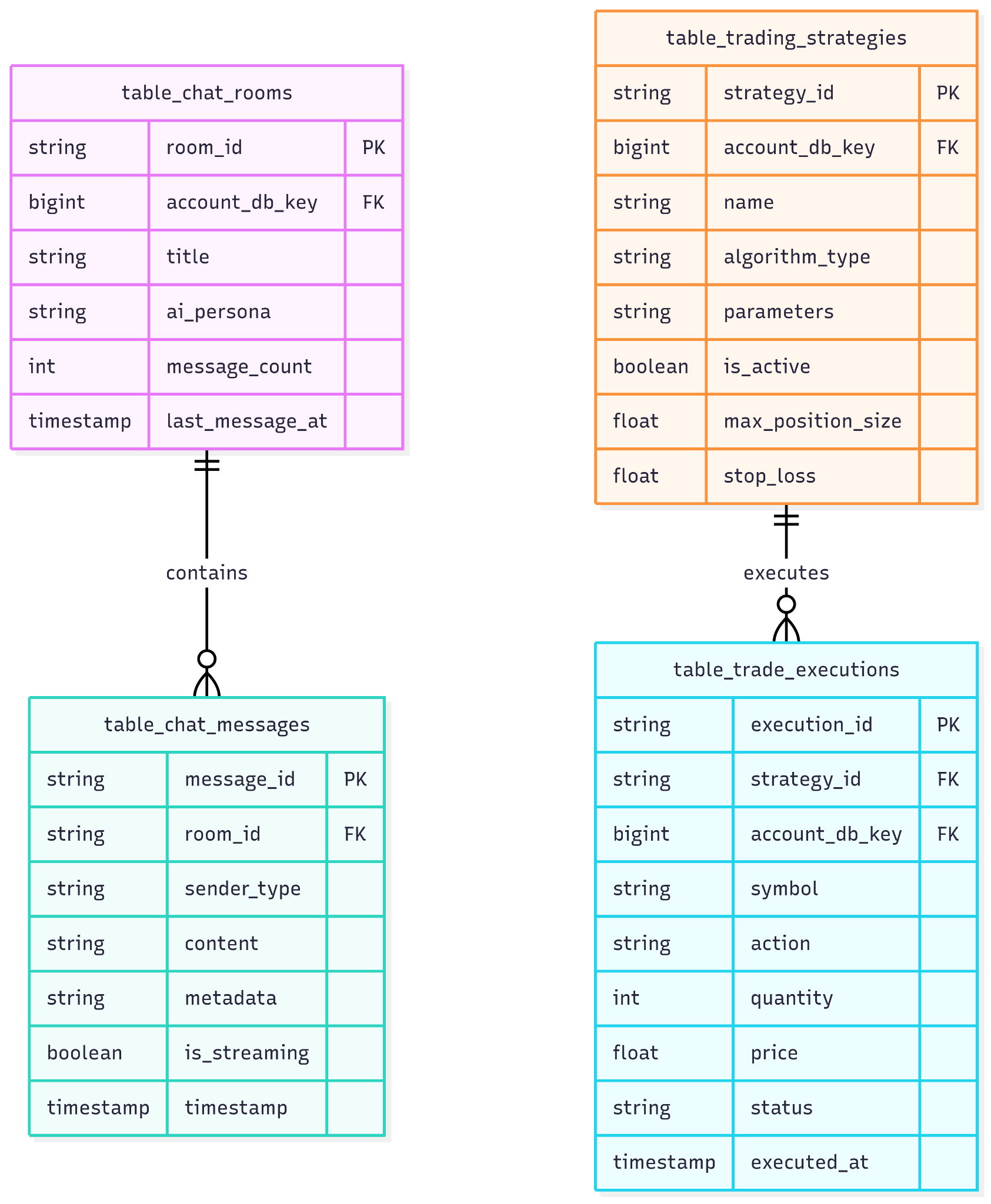


**주요 특징:**

* 계좌 중심의 데이터 구조
* 포트폴리오 성과 일별 추적
* 거래 내역 완전 기록
* 주문 상태 실시간 관리

### 2.5 샤드 DB ERD - AI 채팅 및 자동매매 도메인

AI 기반 투자 조언과 자동매매 전략을 관리하는 테이블들입니다.

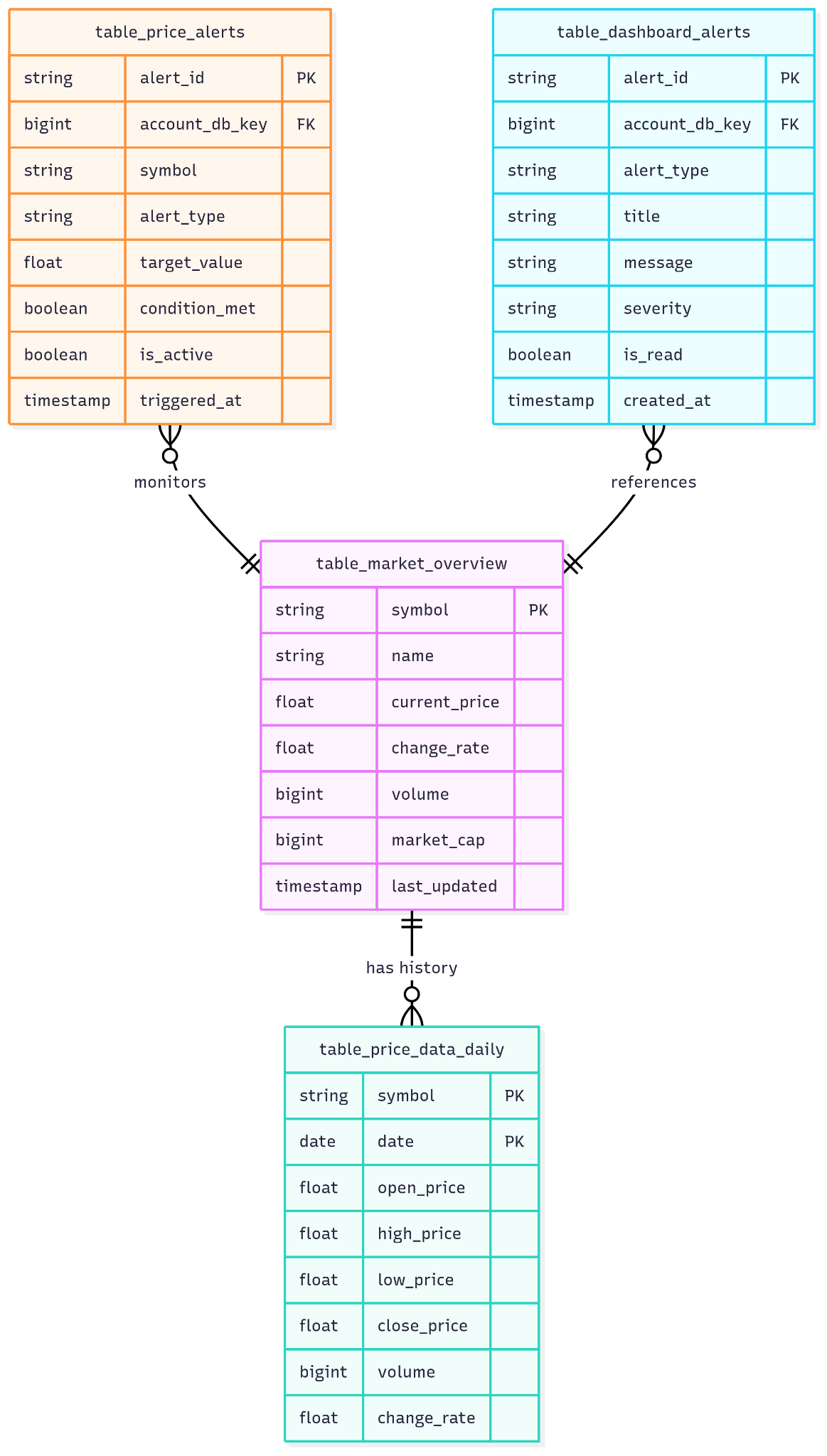


**주요 특징:**

* AI 페르소나별 대화 관리
* 전략별 거래 실행 추적
* 메타데이터를 통한 확장성
* 실시간 스트리밍 지원

### 2.6 샤드 DB ERD - 알림 및 시장 데이터 도메인

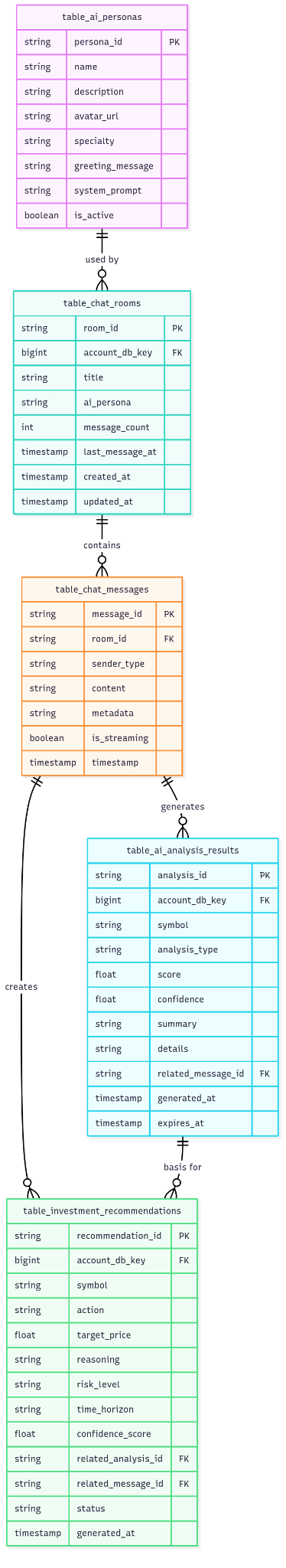
사용자 알림과 시장 데이터를 관리하는 테이블들입니다.



**주요 특징:**

* 다양한 알림 조건 지원
* 시장 데이터 실시간 업데이트
* 가격 히스토리 관리
* 심각도별 알림 분류

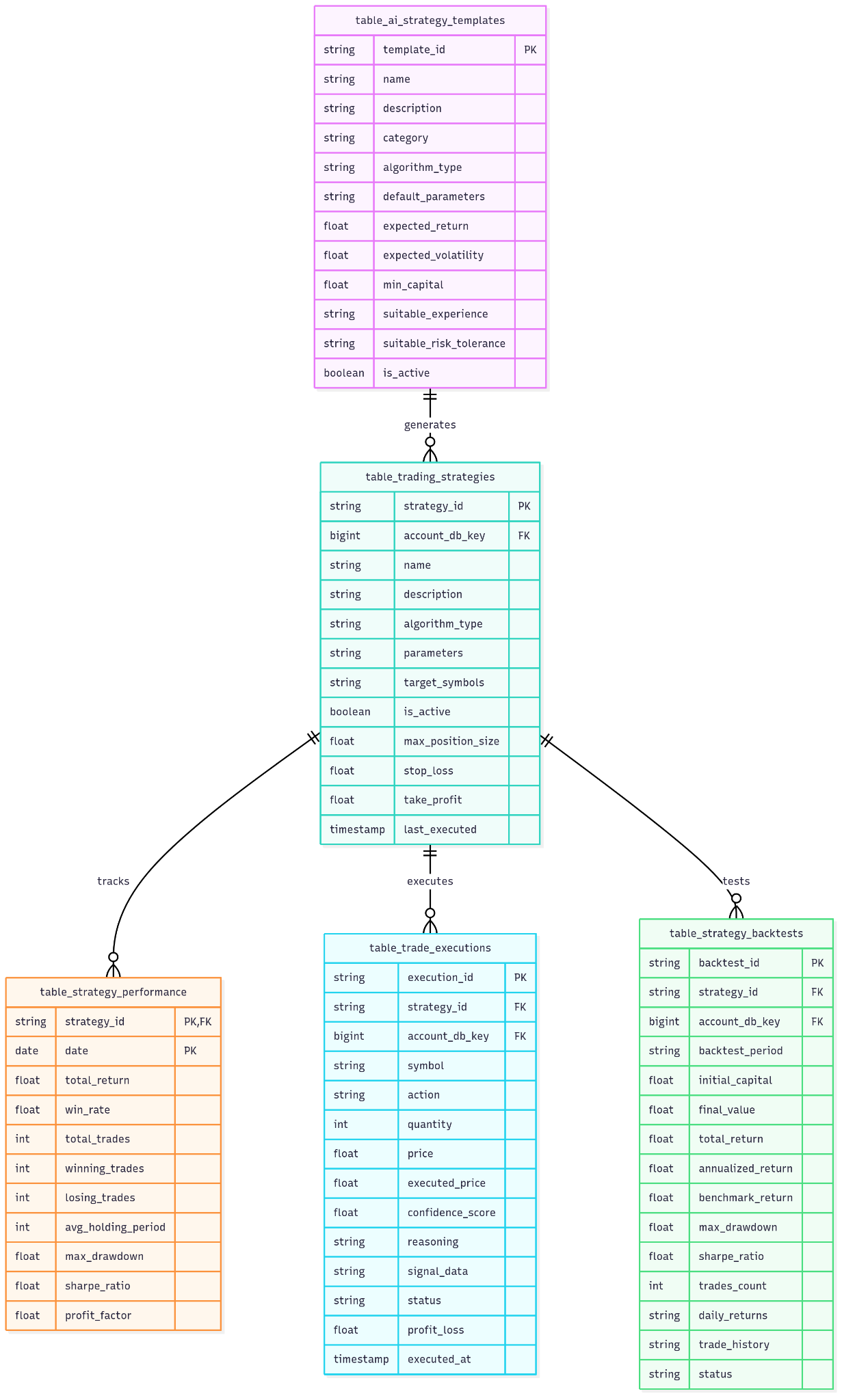
### 2.7 AI 채팅 도메인 상세 ERD



**주요 특징:**

* AI 페르소나별 차별화된 대화 제공
* 메시지 기반 분석 결과 및 투자 추천 생성
* 분석 결과와 추천 간의 연관 관계 추적

### 2.8 자동매매 도메인 상세 ERD



**주요 특징:**

* 전략별 성과 추적 및 백테스트
* AI 템플릿 기반 전략 생성
* 거래 실행 내역 및 손익 관리
* 실시간 전략 성과 모니터링

## 3. 주요 테이블 정의서

### 3.1 글로벌 DB 테이블 (finance\_global)

#### 3.1.1 table\_accountid (계정 정보)

| **컬럼명** | **데이터 타입** | **제약조건** | **기본값** | **설명** |
| --- | --- | --- | --- | --- |
| idx | BIGINT UNSIGNED | PK, AUTO\_INCREMENT | - | 고유 ID |
| platform\_type | TINYINT | NOT NULL | 1 | 플랫폼 유형 |
| account\_id | VARCHAR(100) | NOT NULL, UK (with platform\_type) | - | 계정 ID (이메일) |
| account\_db\_key | BIGINT UNSIGNED | NOT NULL, UK | 0 | 사용자 고유 키 (샤딩 키) |
| account\_status | VARCHAR(15) | NOT NULL | 'Normal' | 계정 상태 |
| block\_endtime | DATETIME | NOT NULL | CURRENT\_TIMESTAMP | 차단 종료시간 |
| is\_withdraw | BIT(1) | NOT NULL | b'0' | 탈퇴 여부 |
| withdraw\_cancel\_count | TINYINT | NOT NULL | 0 | 탈퇴 취소 횟수 |
| withdraw\_time | DATETIME | NOT NULL | CURRENT\_TIMESTAMP | 탈퇴 시간 |
| login\_time | DATETIME | NOT NULL | CURRENT\_TIMESTAMP | 마지막 로그인 시간 |
| logout\_time | DATETIME | NOT NULL | CURRENT\_TIMESTAMP | 마지막 로그아웃 시간 |
| create\_time | DATETIME | NOT NULL | CURRENT\_TIMESTAMP | 생성일시 |
| password\_hash | VARCHAR(255) | NOT NULL | - | 비밀번호 해시 |
| nickname | VARCHAR(50) | NOT NULL | - | 닉네임 |
| email | VARCHAR(100) | - | - | 이메일 |
| account\_level | INT | - | 1 | 계정 레벨 |

**인덱스:**

* PRIMARY KEY (idx)
* UNIQUE KEY ix\_accountid\_platform\_accountid (platform\_type, account\_id)
* KEY ix\_accountid\_accountdbkey (account\_db\_key)

#### 3.1.2 table\_user\_shard\_mapping (샤드 매핑)

| **컬럼명** | **데이터 타입** | **제약조건** | **기본값** | **설명** |
| --- | --- | --- | --- | --- |
| account\_db\_key | BIGINT UNSIGNED | PK, FK | - | 사용자 고유 키 |
| shard\_id | INT | NOT NULL | - | 샤드 ID (1 또는 2) |
| assigned\_at | DATETIME | - | CURRENT\_TIMESTAMP | 할당일시 |
| updated\_at | DATETIME | ON UPDATE CURRENT\_TIMESTAMP | CURRENT\_TIMESTAMP | 수정일시 |

**인덱스:**

* PRIMARY KEY (account\_db\_key)
* INDEX idx\_shard\_id (shard\_id)

#### 3.1.3 table\_otp\_info (OTP 정보)

| **컬럼명** | **데이터 타입** | **제약조건** | **기본값** | **설명** |
| --- | --- | --- | --- | --- |
| account\_db\_key | BIGINT UNSIGNED | PK, FK | - | 사용자 고유 키 |
| secret\_key | VARCHAR(255) | NOT NULL | - | OTP 시크릿 키 (암호화 필수) |
| backup\_codes | TEXT | - | - | 백업 코드 (JSON, 암호화) |
| is\_enabled | BIT(1) | NOT NULL | b'0' | 활성화 여부 |
| qr\_code\_url | TEXT | - | - | QR 코드 URL |
| last\_used\_at | DATETIME | - | NULL | 마지막 사용일시 |
| created\_at | DATETIME | - | CURRENT\_TIMESTAMP | 생성일시 |
| updated\_at | DATETIME | ON UPDATE CURRENT\_TIMESTAMP | CURRENT\_TIMESTAMP | 수정일시 |

#### 

#### 3.1.4 table\_email\_verification (이메일 인증)

| **컬럼명** | **데이터 타입** | **제약조건** | **기본값** | **설명** |
| --- | --- | --- | --- | --- |
| id | BIGINT | PK, AUTO\_INCREMENT | - | 고유 ID |
| email | VARCHAR(100) | NOT NULL, UK (with is\_verified) | - | 이메일 |
| verification\_code | VARCHAR(10) | NOT NULL | - | 인증 코드 |
| account\_id | VARCHAR(100) | - | NULL | 계정 ID |
| is\_verified | BIT(1) | NOT NULL | b'0' | 인증 완료 여부 |
| attempts | INT | - | 0 | 시도 횟수 |
| max\_attempts | INT | - | 5 | 최대 시도 횟수 |
| created\_at | DATETIME | - | CURRENT\_TIMESTAMP | 생성일시 |
| expires\_at | DATETIME | NOT NULL | - | 만료일시 |
| verified\_at | DATETIME | - | NULL | 인증 완료일시 |

**인덱스:**

* PRIMARY KEY (id)
* UNIQUE KEY uk\_email\_active (email, is\_verified)
* INDEX idx\_email (email)
* INDEX idx\_expires\_at (expires\_at)
* INDEX idx\_account\_id (account\_id)

#### 

#### 3.1.5 table\_user\_profiles (사용자 프로필)

| **컬럼명** | **데이터 타입** | **제약조건** | **기본값** | **설명** |
| --- | --- | --- | --- | --- |
| account\_db\_key | BIGINT UNSIGNED | PK, FK | - | 사용자 고유 키 |
| investment\_experience | ENUM | - | 'BEGINNER' | 투자 경험 (BEGINNER/INTERMEDIATE/EXPERT) |
| risk\_tolerance | ENUM | - | 'MODERATE' | 위험 감수 성향 (CONSERVATIVE/MODERATE/AGGRESSIVE) |
| investment\_goal | ENUM | - | 'GROWTH' | 투자 목표 (GROWTH/INCOME/PRESERVATION) |
| monthly\_budget | DECIMAL(15,2) | - | 0.00 | 월 투자 예산 |
| profile\_completed | BIT(1) | NOT NULL | b'0' | 프로필 완성 여부 |
| birth\_year | INT | - | NULL | 출생년도 |
| birth\_month | INT | - | NULL | 출생월 |
| birth\_day | INT | - | NULL | 출생일 |
| gender | ENUM | - | NULL | 성별 (M/F/OTHER) |
| country | VARCHAR(10) | - | 'KR' | 국가 코드 |
| timezone | VARCHAR(50) | - | 'Asia/Seoul' | 시간대 |
| created\_at | DATETIME | - | CURRENT\_TIMESTAMP | 생성일시 |
| updated\_at | DATETIME | ON UPDATE CURRENT\_TIMESTAMP | CURRENT\_TIMESTAMP | 수정일시 |

**인덱스:**

* PRIMARY KEY (account\_db\_key)
* INDEX idx\_investment\_experience (investment\_experience)
* INDEX idx\_risk\_tolerance (risk\_tolerance)
* INDEX idx\_profile\_completed (profile\_completed)

### 3.2 샤드 DB 테이블 (finance\_shard\_1/2)

#### 3.2.1 table\_user\_accounts (사용자 계좌)

| **컬럼명** | **데이터 타입** | **제약조건** | **기본값** | **설명** |
| --- | --- | --- | --- | --- |
| account\_db\_key | BIGINT UNSIGNED | PK | - | 사용자 고유 키 |
| account\_number | VARCHAR(20) | NOT NULL, UK | - | 계좌번호 (암호화 권장) |
| balance | DECIMAL(15,2) | - | 0.00 | 잔액 |
| account\_type | ENUM | - | 'checking' | 계좌 유형 (checking/savings/investment) |
| account\_status | ENUM | - | 'active' | 계좌 상태 (active/suspended/closed) |
| currency\_code | VARCHAR(3) | - | 'USD' | 통화 코드 |
| created\_at | DATETIME | - | CURRENT\_TIMESTAMP | 생성일시 |
| updated\_at | DATETIME | ON UPDATE CURRENT\_TIMESTAMP | CURRENT\_TIMESTAMP | 수정일시 |

**인덱스:**

* PRIMARY KEY (account\_db\_key)
* UNIQUE KEY uk\_account\_number (account\_number)
* INDEX idx\_account\_status (account\_status)
* INDEX idx\_updated\_at (updated\_at)

#### 

#### 3.2.2 table\_user\_portfolios (포트폴리오)

| **컬럼명** | **데이터 타입** | **제약조건** | **기본값** | **설명** |
| --- | --- | --- | --- | --- |
| portfolio\_id | BIGINT | PK, AUTO\_INCREMENT | - | 포트폴리오 ID |
| account\_db\_key | BIGINT UNSIGNED | NOT NULL | - | 사용자 고유 키 |
| asset\_code | VARCHAR(10) | NOT NULL, UK (with account\_db\_key) | - | 종목 코드 |
| quantity | DECIMAL(15,6) | - | 0.000000 | 보유 수량 |
| average\_cost | DECIMAL(15,2) | - | 0.00 | 평균 매입가 |
| current\_value | DECIMAL(15,2) | - | 0.00 | 현재 가치 |
| last\_updated | DATETIME | ON UPDATE CURRENT\_TIMESTAMP | CURRENT\_TIMESTAMP | 최종 업데이트 |

**인덱스:**

* PRIMARY KEY (portfolio\_id)
* UNIQUE KEY uk\_user\_asset (account\_db\_key, asset\_code)
* INDEX idx\_account\_db\_key (account\_db\_key)
* INDEX idx\_asset\_code (asset\_code)

#### 

#### 3.2.3 table\_chat\_rooms (AI 채팅방)

| **컬럼명** | **데이터 타입** | **제약조건** | **기본값** | **설명** |
| --- | --- | --- | --- | --- |
| room\_id | VARCHAR(32) | PK | - | 채팅방 ID |
| account\_db\_key | BIGINT UNSIGNED | NOT NULL | - | 사용자 고유 키 |
| title | VARCHAR(200) | - | NULL | 채팅방 제목 |
| ai\_persona | ENUM | - | 'GPT4O' | AI 페르소나 |
| message\_count | INT | - | 0 | 메시지 수 |
| last\_message\_at | DATETIME | - | NULL | 마지막 메시지 시간 |
| created\_at | DATETIME | - | CURRENT\_TIMESTAMP | 생성일시 |
| updated\_at | DATETIME | ON UPDATE CURRENT\_TIMESTAMP | CURRENT\_TIMESTAMP | 수정일시 |

**AI 페르소나 옵션:**

* GPT4O
* WARREN\_BUFFETT
* PETER\_LYNCH
* GIGA\_BUFFETT

**인덱스:**

* PRIMARY KEY (room\_id)
* INDEX idx\_account\_db\_key (account\_db\_key)
* INDEX idx\_ai\_persona (ai\_persona)
* INDEX idx\_created\_at (created\_at)

#### 

#### 3.2.4 table\_trading\_strategies (매매 전략)

| **컬럼명** | **데이터 타입** | **제약조건** | **기본값** | **설명** |
| --- | --- | --- | --- | --- |
| strategy\_id | VARCHAR(50) | PK | - | 전략 ID |
| account\_db\_key | BIGINT UNSIGNED | NOT NULL | - | 사용자 고유 키 |
| name | VARCHAR(200) | NOT NULL | - | 전략명 |
| description | TEXT | - | - | 전략 설명 |
| algorithm\_type | ENUM | - | 'AI\_GENERATED' | 알고리즘 유형 |
| parameters | TEXT | - | - | 전략 파라미터 (JSON) |
| target\_symbols | TEXT | - | - | 대상 종목 (JSON 배열) |
| is\_active | BIT(1) | NOT NULL | b'0' | 활성화 여부 |
| max\_position\_size | DECIMAL(5,4) | - | 0.1000 | 최대 포지션 비율 (10%) |
| stop\_loss | DECIMAL(10,4) | - | -0.0500 | 손절선 (-5%) |
| take\_profit | DECIMAL(10,4) | - | 0.1500 | 익절선 (+15%) |
| created\_at | DATETIME | - | CURRENT\_TIMESTAMP | 생성일시 |
| updated\_at | DATETIME | ON UPDATE CURRENT\_TIMESTAMP | CURRENT\_TIMESTAMP | 수정일시 |
| last\_executed | DATETIME | - | NULL | 마지막 실행일시 |

**알고리즘 유형:**

* MOMENTUM
* MEAN\_REVERSION
* AI\_GENERATED
* RSI
* MACD
* BOLLINGER

**인덱스:**

* PRIMARY KEY (strategy\_id)
* INDEX idx\_account\_db\_key (account\_db\_key)
* INDEX idx\_algorithm\_type (algorithm\_type)
* INDEX idx\_is\_active (is\_active)
* INDEX idx\_created\_at (created\_at)

#### 3.2.5 table\_price\_alerts (가격 알림)

| **컬럼명** | **데이터 타입** | **제약조건** | **기본값** | **설명** |
| --- | --- | --- | --- | --- |
| alert\_id | VARCHAR(32) | PK | - | 알림 ID |
| account\_db\_key | BIGINT UNSIGNED | NOT NULL | - | 사용자 고유 키 |
| symbol | VARCHAR(10) | NOT NULL | - | 종목 코드 |
| alert\_type | ENUM | NOT NULL | - | 알림 유형 |
| target\_value | DECIMAL(15,4) | NOT NULL | - | 목표값 |
| current\_value | DECIMAL(15,4) | - | 0.0000 | 현재값 |
| condition\_met | BIT(1) | NOT NULL | b'0' | 조건 충족 여부 |
| is\_active | BIT(1) | NOT NULL | b'1' | 활성화 여부 |
| notification\_sent | BIT(1) | NOT NULL | b'0' | 알림 전송 여부 |
| message | VARCHAR(500) | - | NULL | 알림 메시지 |
| created\_at | DATETIME | - | CURRENT\_TIMESTAMP | 생성일시 |
| triggered\_at | DATETIME | - | NULL | 트리거 시간 |
| expires\_at | DATETIME | - | NULL | 만료일시 |

**알림 유형:**

* PRICE\_ABOVE
* PRICE\_BELOW
* CHANGE\_RATE\_ABOVE
* CHANGE\_RATE\_BELOW
* VOLUME\_SPIKE

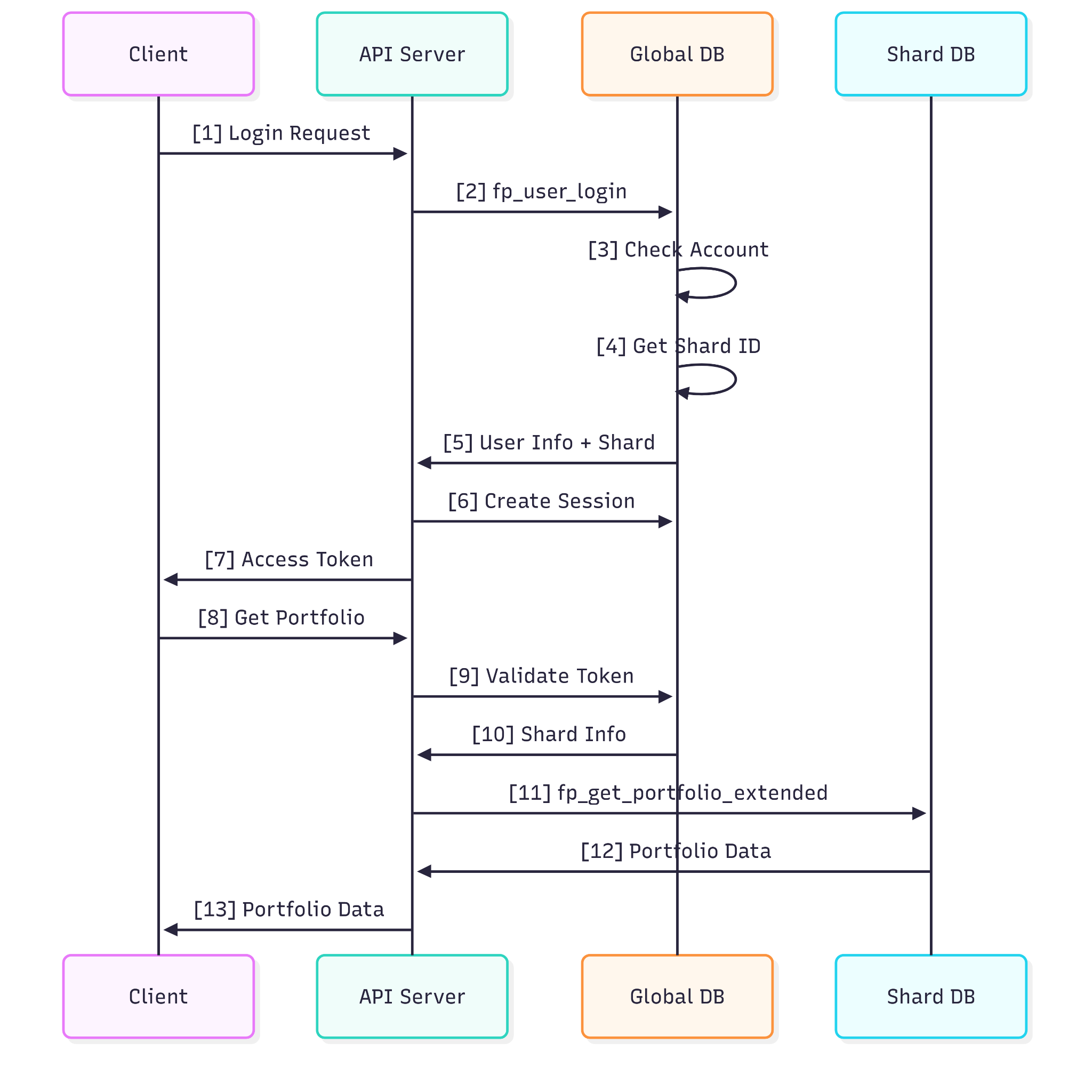
**인덱스:**

* PRIMARY KEY (alert\_id)
* INDEX idx\_account\_db\_key (account\_db\_key)
* INDEX idx\_symbol (symbol)
* INDEX idx\_alert\_type (alert\_type)
* INDEX idx\_is\_active (is\_active)
* INDEX idx\_condition\_met (condition\_met)

## 

## 4. API 흐름도 및 데이터 접근 패턴

### 4.1 사용자 인증 플로우



### 4.2 주요 API 엔드포인트와 프로시저 매핑

| **API 엔드포인트** | **사용 프로시저** | **데이터베이스** | **설명** |
| --- | --- | --- | --- |
| POST /api/account/signup | fp\_user\_register | finance\_global | 회원가입 |
| POST /api/account/login | fp\_user\_login\_with\_otp | finance\_global | 로그인 |
| POST /api/account/otp-setup | fp\_otp\_setup | finance\_global | OTP 설정 |
| GET /api/portfolio | fp\_get\_portfolio\_extended | finance\_shard\_N | 포트폴리오 조회 |
| POST /api/chat/room | fp\_create\_chat\_room | finance\_shard\_N | 채팅방 생성 |
| POST /api/chat/message | fp\_send\_chat\_message | finance\_shard\_N | 메시지 전송 |
| POST /api/autotrade/strategy | fp\_create\_trading\_strategy | finance\_shard\_N | 전략 생성 |
| GET /api/market/price | fp\_get\_price\_data | finance\_shard\_N | 시세 조회 |

## 5. 제약조건 명세

### 5.1 기본키 (Primary Key) 제약조건

-- 모든 테이블은 반드시 기본키를 가져야 함

-- 글로벌 DB: idx 또는 account\_db\_key를 기본키로 사용

-- 샤드 DB: 복합키 또는 고유 ID 사용

-- 예시

PRIMARY KEY (`account\_db\_key`),

PRIMARY KEY (`account\_db\_key`, `symbol`),

PRIMARY KEY (`room\_id`)

### 5.2 외래키 (Foreign Key) 제약조건

-- 샤드 간 외래키는 사용하지 않음 (샤딩 특성상)

-- 동일 DB 내에서만 외래키 사용

-- 글로벌 DB 예시

FOREIGN KEY (`account\_db\_key`) REFERENCES `table\_accountid`(`account\_db\_key`) ON DELETE CASCADE

-- 샤드 DB 예시

FOREIGN KEY (`room\_id`) REFERENCES `table\_chat\_rooms`(`room\_id`) ON DELETE CASCADE

### 5.3 유니크 제약조건

-- 중복 방지를 위한 유니크 인덱스

UNIQUE KEY `ix\_accountid\_platform\_accountid` (`platform\_type`,`account\_id`),

UNIQUE KEY `uk\_account\_number` (`account\_number`),

UNIQUE KEY `uk\_user\_asset` (`account\_db\_key`,`asset\_code`)

### 5.4 체크 제약조건 (애플리케이션 레벨에서 처리)

* 잔액은 0 이상이어야 함
* 수량은 0 이상이어야 함
* 날짜는 유효한 범위 내여야 함

## 6. 쿼리 성능 고려사항

### 6.1 인덱싱 전략

-- 자주 조회되는 컬럼에 인덱스 생성

INDEX `idx\_account\_db\_key` (`account\_db\_key`),

INDEX `idx\_symbol` (`symbol`),

INDEX `idx\_created\_at` (`created\_at`),

INDEX `idx\_status` (`status`),

INDEX `idx\_is\_active` (`is\_active`)

### 6.2 샤딩 전략

* **샤딩 키**: account\_db\_key (사용자 고유 키)
* **샤드 할당**: (account\_db\_key % 2) + 1
* **샤드 수**: 초기 2개, 필요시 확장 가능
* **데이터 분산**: 사용자별로 독립적인 데이터 저장

### 6.3 스토어드 프로시저 사용

* 모든 데이터 접근은 스토어드 프로시저를 통해 수행
* 네트워크 라운드트립 최소화
* SQL 인젝션 방지
* 일관된 에러 처리

### 6.4 파티셔닝 고려사항

-- 시계열 데이터는 날짜별 파티셔닝 고려

-- 예: table\_portfolio\_performance, table\_price\_data\_daily

PARTITION BY RANGE (TO\_DAYS(date)) (

PARTITION p2024 VALUES LESS THAN (TO\_DAYS('2025-01-01')),

PARTITION p2025 VALUES LESS THAN (TO\_DAYS('2026-01-01'))

);

## 7. 데이터 보안 고려사항

### 7.1 개인정보 암호화

-- 민감한 개인정보 필드 암호화

-- 암호화 대상 필드:

-- - account\_id (이메일)

-- - email

-- - account\_number

-- - secret\_key (OTP)

-- - backup\_codes

-- 암호화 예시 (애플리케이션 레벨)

INSERT INTO table\_accountid (account\_id, email) VALUES

(AES\_ENCRYPT('user@example.com', 'encryption\_key'),

AES\_ENCRYPT('user@example.com', 'encryption\_key'));

### 7.2 비밀번호 보안

* SHA256 해시 사용
* Salt 추가 권장
* 평문 저장 금지

### 7.3 접근 제어

-- 사용자별 데이터 격리

-- 모든 쿼리에 account\_db\_key 조건 필수

WHERE account\_db\_key = p\_account\_db\_key

-- 권한 체크 예시

IF v\_account\_db\_key != p\_account\_db\_key THEN

SIGNAL SQLSTATE '45000' SET MESSAGE\_TEXT = 'Access denied';

END IF;

### 7.4 감사 로깅

-- 모든 에러와 중요 작업 로깅

INSERT INTO table\_errorlog (procedure\_name, error\_state, error\_no, error\_message, param)

VALUES ('fp\_user\_login', @ErrorState, @ErrorNo, @ErrorMessage, ProcParam);

-- 설정 변경 이력 추적

INSERT INTO table\_settings\_history (account\_db\_key, setting\_category, setting\_key, old\_value, new\_value)

VALUES (p\_account\_db\_key, 'SECURITY', 'password\_change', NULL, NOW());

### 7.5 데이터 마스킹

-- 개발/테스트 환경에서 민감 데이터 마스킹

-- 예시: 계좌번호 마스킹

SELECT

CONCAT(LEFT(account\_number, 4), '\*\*\*\*', RIGHT(account\_number, 4)) as masked\_account\_number

FROM table\_user\_accounts;

## 8. 백업 및 복구 전략

### 8.1 백업 정책

* **전체 백업**: 주 1회 (일요일 새벽)
* **증분 백업**: 일 1회
* **트랜잭션 로그 백업**: 1시간마다
* **보관 기간**: 30일

### 8.2 복구 시나리오

1. **Point-in-Time Recovery**: 특정 시점으로 복구
2. **샤드별 복구**: 개별 샤드 단위 복구 가능
3. **재해 복구**: 다른 리전으로 페일오버

## 9. 모니터링 및 유지보수

### 9.1 모니터링 지표

* 쿼리 응답 시간
* 샤드별 부하 분산
* 스토리지 사용률
* 커넥션 풀 상태

### 9.2 정기 유지보수

* 인덱스 리빌드: 월 1회
* 통계 업데이트: 주 1회
* 로그 테이블 정리: 일 1회
* 샤드 리밸런싱: 분기 1회

## 10. 확장성 고려사항

### 10.1 수평적 확장 (Scale-out)

* 샤드 추가를 통한 확장
* 읽기 전용 복제본 활용
* 캐싱 레이어 도입 (Redis)

### 10.2 수직적 확장 (Scale-up)

* 하드웨어 스펙 업그레이드
* 스토리지 용량 증설
* 메모리 확장

### 10.3 장애 대응

* 샤드별 독립적 운영으로 장애 격리
* 마스터-슬레이브 구조로 고가용성 확보
* 자동 페일오버 설정