ERD: 가족 질문 생성/배포/분석 (with GPT)

아래 다이어그램은 **템플릿 / 인스턴스 / 배포(할당)** 를 분리하고, GPT API를 통한 **질문 생성 메타데이터**와 **답변 분석 메타데이터**를 포함합니다. Mermaid를 지원하는 뷰어나 VSCode 확장(Markdown Preview Mermaid Support 등)에서 렌더링하면 시각화됩니다.

```
erDiagram
    FAMILY ||--o{ MEMBER : has
    MEMBER ||--o| MEMBER_PROFILE : has
    MEMBER ||--o{ MEMBER_AFFINITY : relates

    FAMILY ||--o{ QUESTION_INSTANCE : has
    QUESTION_TEMPLATE ||--o{ QUESTION_INSTANCE : instantiates
    QUESTION_INSTANCE ||--o{ QUESTION_ASSIGNMENT : distributes
    MEMBER ||--o{ QUESTION_ASSIGNMENT : receives

    QUESTION_ASSIGNMENT ||--o{ ANSWER : has
    ANSWER ||--o{ ANSWER_ANALYSIS : produces
    ANSWER ||--o{ COMMENT : has
    MEMBER ||--o{ COMMENT : writes

    FAMILY {
      bigint id PK
      text   name
      timestamptz created_at
    }

    MEMBER {
      bigint id PK
      bigint family_id FK
      text   name
      text   role
      date   birthdate
      timestamptz created_at
    }

    MEMBER_PROFILE {
      bigint member_id PK, FK
      jsonb  preferences      "주제/톤/금기 등"
      jsonb  engagement_stats "응답률/길이/감정 평균 등"
      timestamptz last_ai_update_at
    }

    MEMBER_AFFINITY {
      bigint id PK
      bigint subject_member_id FK
      bigint target_member_id FK
```

```
  float  affinity_score
  timestamptz last_updated_at
  unique(subject_member_id, target_member_id)
}

QUESTION_TEMPLATE {
  bigint id PK
  text   content           "예: {subject}와의 최애 추억?"
  text   category
  jsonb  tags
  boolean subject_required
  text   reuse_scope       "global|per_family|per_subject"
  int    cooldown_days
  text   language
  text   tone
  timestamptz created_at
}

QUESTION_INSTANCE {
  bigint id PK
  bigint family_id FK
  bigint template_id FK NULL
  bigint subject_member_id FK NULL
  text   content                "치환된 최종 질문(고정)"
  date   planned_date
  text   status                 "draft|scheduled|sent|canceled"
  text   generated_by           "ai|manual"
  text   generation_model
  jsonb  generation_parameters
  text   generation_prompt
  jsonb  generation_metadata
  float  generation_confidence
  timestamptz generated_at
  unique(family_id, planned_date)
}

QUESTION_ASSIGNMENT {
  bigint id PK
  bigint instance_id FK
  bigint recipient_member_id FK
  timestamptz due_at
  timestamptz sent_at
  text   state                  "pending|answered|expired"
  timestamptz last_delivered_at
  unique(instance_id, recipient_member_id)
}

ANSWER {
  bigint id PK
```

```
    bigint assignment_id FK
    bigint author_member_id FK
    text   content
    timestamptz created_at
  }

  ANSWER_ANALYSIS {
    bigint id PK
    bigint answer_id FK
    text   analysis_model
    jsonb  analysis_parameters
    text   analysis_prompt
    text   analysis_raw          "원문 보관 (선택)"
    text   analysis_version
    text   summary
    jsonb  categories            "감정/주제 등"
    jsonb  scores                "선호/유사도 점수 등"
    timestamptz created_at
  }

  COMMENT {
    bigint id PK
    bigint answer_id FK
    bigint commenter_member_id FK
    text   content
    timestamptz created_at
  }
```

관계 & 제약 정리

- **가족당 하루 1문**: `QUESTION_INSTANCE(unique(family_id, planned_date))`
- **중복 배포 방지**: `QUESTION_ASSIGNMENT(unique(instance_id, recipient_member_id))`
- **호감도/관계 점수**: `MEMBER_AFFINITY(subject_member_id, target_member_id)` 유니크
- **질문 텍스트는 인스턴스에만 고정 저장** → 템플릿 수정이 과거 기록에 영향 없음
- **답변은 배포 기준으로 연결** → `ANSWER.assignment_id` (누구에게 배포된 질문의 답인지 명확)

쿨다운/재사용 로직 포인트

- 템플릿 재사용 범위(`reuse_scope`)와 `cooldown_days`를 사용해 가족/주인공 단위 제한
- 인스턴스 생성 시 프롬프트/모델/파라미터를 함께 기록하여 추적성 확보

개인화 데이터 흐름

1. 프로필/관계(`MEMBER_PROFILE`, `MEMBER_AFFINITY`)를 프롬프트에 주입 → GPT 개인화 생성
2. 답변 수집 후 `ANSWER_ANALYSIS` 저장 → 배치/트리거로 프로필/관계 갱신

샘플 DDL 스니펫 (PostgreSQL)

```
-- 핵심 테이블 일부만 발췌
CREATE TABLE question_template (
 id BIGSERIAL PRIMARY KEY,
 content TEXT NOT NULL,
 category TEXT,
 tags JSONB,
 subject_required BOOLEAN DEFAULT FALSE,
 reuse_scope TEXT DEFAULT 'per_family',
 cooldown_days INT DEFAULT 60,
 created_at TIMESTAMPTZ DEFAULT now()
);

CREATE TABLE question_instance (
 id BIGSERIAL PRIMARY KEY,
 family_id BIGINT NOT NULL REFERENCES family(id),
 template_id BIGINT REFERENCES question_template(id),
 subject_member_id BIGINT REFERENCES member(id),
 content TEXT NOT NULL,
 planned_date DATE NOT NULL,
 status TEXT DEFAULT 'scheduled',
 generated_by TEXT DEFAULT 'ai',
 generation_model TEXT,
 generation_parameters JSONB,
 generation_prompt TEXT,
 generation_metadata JSONB,
 generation_confidence DOUBLE PRECISION,
 generated_at TIMESTAMPTZ DEFAULT now(),
 UNIQUE(family_id, planned_date)
);

CREATE TABLE question_assignment (
 id BIGSERIAL PRIMARY KEY,
 instance_id BIGINT NOT NULL REFERENCES question_instance(id),
 recipient_member_id BIGINT NOT NULL REFERENCES member(id),
 due_at TIMESTAMPTZ,
 sent_at TIMESTAMPTZ,
 state TEXT DEFAULT 'pending',
 last_delivered_at TIMESTAMPTZ,
 UNIQUE(instance_id, recipient_member_id)
);

CREATE TABLE answer (
 id BIGSERIAL PRIMARY KEY,
 assignment_id BIGINT NOT NULL REFERENCES question_assignment(id),
 author_member_id BIGINT NOT NULL REFERENCES member(id),
 content TEXT NOT NULL,
 created_at TIMESTAMPTZ DEFAULT now()
);
```

```
CREATE TABLE answer_analysis (
  id BIGSERIAL PRIMARY KEY,
  answer_id BIGINT NOT NULL REFERENCES answer(id),
  analysis_model TEXT,
  analysis_parameters JSONB,
  analysis_prompt TEXT,
  analysis_raw TEXT,
  analysis_version TEXT,
  summary TEXT,
  categories JSONB,
  scores JSONB,
  created_at TIMESTAMPTZ DEFAULT now()
);
```

필요하면 **쿨다운 선택 쿼리**, **프로필 갱신 트리거**(예: 답변 저장 시 engagement_stats 업데이트)까지 이어서 추가해드릴게요.