

Project Deliverable 2

第18小组 刘志锋 黄柏曛 时啟轩

一、阶段目标

- 将第1阶段的 ER 模型无歧义地映射为关系模式，形成可实现、可检查约束的逻辑设计。
- 明确各表的主键、外键、候选键；补充参照完整性之外的业务约束与可实施策略。
- 为后续物理设计（范式化、索引、分区）与实现（DDL/迁移、数据访问层）提供基础。

二、对阶段一的修改

- 未修改概念结构与边界条件；仅在实现细节上“落地化”：
 - 将复合属性 `User.name` 物化为 `first_name`、`last_name` 两列（便于查询与排序）。
 - 将特化/泛化采用“类表继承（Class Table Inheritance）”：
`Doctor / Specialist / Therapist` 以 `license_number` 作为 PK 兼 FK 指向 `Provider`。
 - 将派生属性以“生成列/视图/物化列”三选一策略表达（具体在约束章节阐明）。

三、ER → 关系模型映射说明

遵循标准映射算法：实体→表；关系→外键/连接表；特化→类表继承；多值/复合属性→拆分列或独立表。

3.1 实体到关系表

- User(health_id PK, first_name, last_name)
- ContactEmail(email PK, is_verified, user_health_id FK nullable)
- Phone(phone_number PK, is_verified, user_health_id FK nullable)
- Provider(license_number PK, name, is_verified)
- Doctor(license_number PK/FK→Provider)
- Specialist(license_number PK/FK→Provider)
- Therapist(license_number PK/FK→Provider)
- FamilyGroup(group_id PK, name)
- MonthlyReport(report_id PK, user_health_id FK, month, summary, steps_total 派生)
- HealthMetric(metric_id PK, user_health_id FK, metric_date, metric_type, metric_value)
- Challenge(challenge_id PK, creator_user_health_id FK, goal, start_date, end_date)

3.2 关系到连接表与外键

- User—ContactEmail: 0..N 邮箱/邮箱→用户 0..1 ⇒ `contactEmail.user_health_id` 可空 FK。
- User—Phone: 用户 0..1 电话/电话→用户 0..1 ⇒ `Phone.user_health_id` 可空 FK，且对 `user_health_id` 施加唯一约束（确保“每用户至多一条电话记录”）。

- User—Provider 多对多 (含属性 `isPrimaryCare`, `linkedAt`) \Rightarrow
`UserProvider(user_provider_id PK, user_health_id FK, provider_license_number FK, is_primary_care, linked_at)`; 候选键 `(user_health_id, provider_license_number)`; 并对 `is_primary_care=TRUE` 施加条件唯一 (每用户最多一位主治)。
- FamilyGroup—User 多对多 (含 `role`, `joined_at`) \Rightarrow `FamilyMembership(membership_id PK, group_id FK, user_health_id FK, role, joined_at)`; 候选键 `(group_id, user_health_id)`。
- User—User 递归 (guardian/dependent) \Rightarrow `UserDelegation(delegation_id PK, guardian_user_health_id FK, dependent_user_health_id FK, created_at)`; 候选键 `(guardian_user_health_id, dependent_user_health_id)`; 并约束 `guardian ≠ dependent`。
- Appointment 用户预约提供者 \Rightarrow `Appointment(appointment_id PK, user_health_id FK, provider_license_number FK, date_time, type, memo, status, cancel_reason)`; 候选键 `(user_health_id, provider_license_number, date_time)`; 取消窗口属业务约束 (见下)。
- User—Challenge (creates) \Rightarrow `Challenge.creator_user_health_id` 非空 FK。
- Challenge—User (participates) \Rightarrow `ChallengeParticipation(participation_id PK, challenge_id FK, user_health_id FK, progress_value, status, updated_at)`; 候选键 `(challenge_id, user_health_id)`。
- Invitation (User sends, 可选关联 Challenge) \Rightarrow `Invitation(invitation_id PK, sender_user_health_id FK, challenge_id FK NULL, type, target_email, target_phone, created_at, completed_at, expires_at 派生, status)`; 约束“目标至少其一 (email/phone) 存在”。
- User—MonthlyReport/HealthMetric: 一对多 \Rightarrow 各自带 `user_health_id` 外键。

3.3 特化/泛化映射

- 采用类表继承: 子类表 PK 即父类表 PK, 且为 FK 指向父类 (删除父类时级联删除子类)。
- 可保留父表共享属性 (如 `name`, `is_verified`)。

3.4 属性、枚举与派生

- 枚举以 CHECK 约束实现, 兼容多数 SQL 方言; 必要时集中放置“字典表”。
- 派生字段 (如 `Invitation.expires_at = created_at + 15 days`、`MonthlyReport.steps_total`) 可:
 1. 运行时视图计算;
 2. 生成列 (方言依赖, 如 PostgreSQL 计算列/触发器);
 3. 物化后异步维护 (批任务/触发器)。

四、关系模式清单 (主键/外键摘要)

(完整 DDL 见 `schema.sql`)

- User(`health_id` PK)
- ContactEmail(`email` PK, `user_health_id` FK \rightarrow User)

- Phone(phone_number PK, user_health_id FK→User, UNIQUE(user_health_id))
- Provider(license_number PK)
- Doctor(license_number PK/FK→Provider)
- Specialist(license_number PK/FK→Provider)
- Therapist(license_number PK/FK→Provider)
- UserProvider(user_provider_id PK, user_health_id FK→User, provider_license_number FK→Provider, UNIQUE(user_health_id, provider_license_number))
- FamilyGroup(group_id PK)
- FamilyMembership(membership_id PK, group_id FK→FamilyGroup, user_health_id FK→User, UNIQUE(group_id, user_health_id))
- UserDelegation(delegation_id PK, guardian_user_health_id FK→User, dependent_user_health_id FK→User)
- Appointment(appointment_id PK, user_health_id FK→User, provider_license_number FK→Provider)
- Challenge(challenge_id PK, creator_user_health_id FK→User)
- ChallengeParticipation(participation_id PK, challenge_id FK→Challenge, user_health_id FK→User, UNIQUE(challenge_id, user_health_id))
- Invitation(invitation_id PK, sender_user_health_id FK→User, challenge_id FK→Challenge NULLABLE)
- MonthlyReport(report_id PK, user_health_id FK→User)
- HealthMetric(metric_id PK, user_health_id FK→User)

五、候选键

- User: 无 (`health_id` 为全局标识)。
- ContactEmail: 无 (`email` 为主键; 每个 `email` 至多关联一个用户)。
- Phone: 无 (`phone_number` 为主键; 对 `user_health_id` 施加唯一以满足“每用户至多一部电话”)。
- Provider: 无 (`license_number` 为全局标识)。
- Doctor / Specialist / Therapist: 继承 `Provider.license_number`。
- UserProvider: `(user_health_id, provider_license_number)`。
- FamilyGroup: 无 (允许同名)。
- FamilyMembership: `(group_id, user_health_id)`。
- UserDelegation: `(guardian_user_health_id, dependent_user_health_id)`。
- Appointment: `(user_health_id, provider_license_number, date_time)`。
- Challenge: 无。
- ChallengeParticipation: `(challenge_id, user_health_id)`。

- Invitation: 无（目标与时间不组成强唯一）。
- MonthlyReport: `(user_health_id, month)`。
- HealthMetric: 可选 `(user_health_id, metric_type, metric_date)` (若限定“同一用户/类型/日期唯一”)。

六、参照完整性之外的约束

- Primary Care 唯一：同一用户 `is_primary_care=TRUE` 至多一条（条件唯一索引/触发器实现）。
- 预约取消窗口：仅允许在预约时间前 ≥ 24 小时取消；取消须记录 `cancel_reason`（应用层校验 + 触发器约束状态流转）。
- 邀请有效期：`expires_at = created_at + 15 days`；过期自动转为 `Expired`，完成后不可再接受（定时任务/触发器）。
- 验证要求：未验证的 `email/phone` 不得用于关键动作；未验证 `Provider` 不可设为主治（应用层 + 触发器）。
- 递归约束：`guardian_user_health_id ≠ dependent_user_health_id`；同一对主体不得重复授权。
- 挑战时间：`end_date >= start_date`。
- 邀请目标：`target_email` 与 `target_phone` 至少其一非空。

七、困难与解决方案

- 条件唯一（主治唯一）：不同数据库对“部分唯一索引”支持差异较大。方案：PostgreSQL 用 `UNIQUE WHERE is_primary_care`；其他方言用触发器或在应用层 + 定时稽核。
- 枚举可移植性：不同方言对 ENUM 支持不一。方案：统一用 `CHECK` 约束或字典表，避免硬 ENUM。
- 派生字段一致性：`expires_at`、`steps_total` 的来源多样。方案：提供三种实现路径；在线路径优先视图/生成列，离线路径用批量刷新物化视图。
- 递归关系与删除策略：避免误级联导致历史丢失。方案：递归表不级联删除用户，改“限制删除”或逻辑删除用户。