

# 数据库结构文档

医生人格特质标注系统的数据库结构说明。

## 数据库概览

本系统使用PostgreSQL作为主数据库，包含以下主要表：

- `physicians` - 医生信息表
- `reviews` - 患者评论表
- `tasks` - 标注任务表
- `model_annotations` - 模型标注结果表
- `human_annotations` - 人工标注结果表
- `machine_annotation_evaluations` - 人类对机器标注的评估表

## 表结构详解

### 1. `physicians` — 医生信息表

存储医生的基本信息和简介。

字段名	类型	描述	示例
<code>id</code>	SERIAL	主键	1
<code>phy_id</code>	BIGINT	医生原始数据唯一ID	100745676
<code>npi</code>	BIGINT	国家提供者编号	1659371102
<code>first_name</code>	TEXT	医生名字	ALINA
<code>last_name</code>	TEXT	医生姓氏	GRIGORE
<code>gender</code>	TEXT	性别	F
<code>credential</code>	TEXT	医生头衔（如MD）	MD
<code>specialty</code>	TEXT	医生专业领域	Anesthesiology Physician
<code>practice_zip5</code>	TEXT	实践邮编	89134
<code>business_zip5</code>	TEXT	商业邮编	25304
<code>biography_doc</code>	TEXT	简介（支持HTML格式）	<code>&lt;p&gt;Dr. Grigore is...&lt;/p&gt;</code>
<code>education_doc</code>	TEXT	教育经历（XML格式）	<code>&lt;education&gt;Duke University...&lt;/education&gt;</code>

字段名	类型	描述	示例
num_reviews	INTEGER	评论数量	7
doc_name	TEXT	医生显示名称	Dr. Alina Grigore
zip3	TEXT	邮编前 3 位	891
zip2	TEXT	邮编前 2 位	89
zipcode	TEXT	完整邮编	89134
state	TEXT	所在州	NV
region	TEXT	地区分布	Mountain

特殊说明：

- `biography_doc`: 支持HTML标签，前端会进行渲染
- `education_doc`: 使用`<education>`XML标签格式，前端会自动解析

## 2. reviews — 患者评论表

存储患者对医生的评论信息。

字段名	类型	描述	示例
id	SERIAL	主键	223
physician_id	INTEGER	外键，关联 physicians	10
review_index	INTEGER	评论编号（#0, #1 ...）	0
source	TEXT	来源（Vitals、HG等）	Vitals
date	TIMESTAMP	时间戳	2025-06-05 23:30:18
text	TEXT	评论内容	I had an excellent experience...

索引建议：

```
CREATE INDEX idx_reviews_physician_id ON reviews(physician_id);
CREATE INDEX idx_reviews_date ON reviews(date);
```

## 3. tasks — 标注任务表

管理标注任务的分配和状态。

字段名	类型	描述	示例
id	SERIAL	主键	1

字段名	类型	描述	示例
physician_id	INTEGER	外键，关联 physicians	10
assigned_to	TEXT	分配给的标注者	user001
status	TEXT	任务状态	in_progress
created_at	TIMESTAMP	创建时间	2025-06-05 23:30:18
updated_at	TIMESTAMP	更新时间	2025-06-05 23:35:20

状态枚举：

- pending - 待处理
- in\_progress - 进行中
- completed - 已完成
- cancelled - 已取消

#### 4. model\_annotations — 模型标注结果表

存储AI模型对医生人格特质的分析结果。

字段名	类型	描述	示例
id	SERIAL	主键	1
physician_id	INTEGER	外键，关联 physicians	10
model_name	TEXT	模型名称	GPT-4
trait	TEXT	人格维度	Openness
score	TEXT	打分结果	High
consistency	TEXT	模型一致性描述	Very Consistent
sufficiency	TEXT	模型证据充分性描述	Sufficient
evidence	TEXT	模型提供的原始证据文本	Based on the reviews...

人格特质枚举：

- Openness - 开放性
- Conscientiousness - 尽责性
- Extraversion - 外向性
- Agreeableness - 宜人性
- Neuroticism - 神经质

评分枚举：

- Low - 低
- Moderate - 中等

- High - 高

### 5. human\_annotations — 人工标注结果表

存储人类标注者的标注结果。

字段名	类型	描述	示例
id	SERIAL	主键	1
physician_id	INTEGER	外键，关联 physicians	10
evaluator	TEXT	标注者用户名	user001
task_id	INTEGER	外键，关联 tasks	1
trait	TEXT	人格维度	Openness
score	TEXT	打分结果	High
consistency	TEXT	一致性评估	Very Consistent
sufficiency	TEXT	证据充分性评估	Sufficient
evidence	TEXT	标注者提供的证据文本	The patient reviews show...
timestamp	TIMESTAMP	标注时间	2025-06-05 23:30:18

### 6. machine\_annotation\_evaluations — 机器标注评估表

存储人类标注者对AI模型输出的评价。

字段名	类型	描述	示例
id	SERIAL	主键	1
model_annotation_id	INTEGER	外键，关联 model_annotations	1
evaluator	TEXT	评估者用户名	user001
task_id	INTEGER	外键，关联 tasks	1
ranking	INTEGER	模型排名（1最好）	1
accuracy_score	TEXT	准确性评价	Good
comment	TEXT	主观评价文字	This model provides accurate...
timestamp	TIMESTAMP	评估时间	2025-06-05 23:30:18

准确性评分枚举：

- Excellent - 优秀

- Good - 良好
- Fair - 一般
- Poor - 较差

---

## 关系图

```
physicians (1) ↔ (N) reviews
      ↓
tasks (1) ↔ (N) human_annotations
      ↓
physicians (1) ↔ (N) model_annotations
      ↓
model_annotations (1) ↔ (N) machine_annotation_evaluations
```

---

## 数据库初始化

### 创建数据库

```
CREATE DATABASE physicians;
\c physicians;
```

### 创建表结构

```
-- 医生信息表
CREATE TABLE physicians (
    id SERIAL PRIMARY KEY,
    phy_id BIGINT,
    npi BIGINT UNIQUE,
    first_name TEXT,
    last_name TEXT,
    gender TEXT,
    credential TEXT,
    specialty TEXT,
    practice_zip5 TEXT,
    business_zip5 TEXT,
    biography_doc TEXT,
    education_doc TEXT,
    num_reviews INTEGER,
    doc_name TEXT,
    zip3 TEXT,
    zip2 TEXT,
    zipcode TEXT,
    state TEXT,
    region TEXT
```

```

);

-- 评论表
CREATE TABLE reviews (
    id SERIAL PRIMARY KEY,
    physician_id INTEGER REFERENCES physicians(id),
    review_index INTEGER,
    source TEXT,
    date TIMESTAMP,
    text TEXT
);

-- 任务表
CREATE TABLE tasks (
    id SERIAL PRIMARY KEY,
    physician_id INTEGER REFERENCES physicians(id),
    assigned_to TEXT,
    status TEXT DEFAULT 'pending',
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

-- 模型标注表
CREATE TABLE model_annotations (
    id SERIAL PRIMARY KEY,
    physician_id INTEGER REFERENCES physicians(id),
    model_name TEXT,
    trait TEXT,
    score TEXT,
    consistency TEXT,
    sufficiency TEXT,
    evidence TEXT
);

-- 人工标注表
CREATE TABLE human_annotations (
    id SERIAL PRIMARY KEY,
    physician_id INTEGER REFERENCES physicians(id),
    evaluator TEXT,
    task_id INTEGER REFERENCES tasks(id),
    trait TEXT,
    score TEXT,
    consistency TEXT,
    sufficiency TEXT,
    evidence TEXT,
    timestamp TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

-- 机器标注评估表
CREATE TABLE machine_annotation_evaluations (
    id SERIAL PRIMARY KEY,
    model_annotation_id INTEGER REFERENCES model_annotations(id),
    evaluator TEXT,

```

```
task_id INTEGER REFERENCES tasks(id),
ranking INTEGER,
accuracy_score TEXT,
comment TEXT,
timestamp TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

## 创建索引

```
-- 性能优化索引
CREATE INDEX idx_physicians_npi ON physicians(npi);
CREATE INDEX idx_reviews_physician_id ON reviews(physician_id);
CREATE INDEX idx_tasks_physician_id ON tasks(physician_id);
CREATE INDEX idx_tasks_assigned_to ON tasks(assigned_to);
CREATE INDEX idx_human_annotations_physician_id ON
human_annotations(physician_id);
CREATE INDEX idx_human_annotations_evaluator ON
human_annotations(evaluator);
CREATE INDEX idx_model_annotations_physician_id ON
model_annotations(physician_id);
CREATE INDEX idx_machine_evaluations_model_annotation_id ON
machine_annotation_evaluations(model_annotation_id);
```

---

## 数据导入

### 使用Go导入工具

```
cd backend/cmd/import
go run main.go
```

### 使用Python ETL脚本

```
cd database
python etl.py
```

---

## 查询示例

### 获取医生完整信息

```
SELECT p.*,
       COUNT(r.id) as review_count
```

```
FROM physicians p
LEFT JOIN reviews r ON p.id = r.physician_id
WHERE p.npi = 1659371102
GROUP BY p.id;
```

## 获取标注进度

```
SELECT
    p.doc_name,
    t.assigned_to,
    COUNT(DISTINCT ha.trait) as completed_traits,
    COUNT(DISTINCT ma.trait) as total_traits
FROM physicians p
JOIN tasks t ON p.id = t.physician_id
LEFT JOIN human_annotations ha ON p.id = ha.physician_id
    AND t.id = ha.task_id
LEFT JOIN model_annotations ma ON p.id = ma.physician_id
WHERE t.assigned_to = 'user001'
GROUP BY p.id, p.doc_name, t.assigned_to;
```

## 获取模型评估统计

```
SELECT
    ma.model_name,
    mae.accuracy_score,
    COUNT(*) as evaluation_count,
    AVG(mae.ranking::numeric) as avg_ranking
FROM model_annotations ma
JOIN machine_annotation_evaluations mae ON ma.id =
mae.model_annotation_id
GROUP BY ma.model_name, mae.accuracy_score
ORDER BY ma.model_name, avg_ranking;
```

---

## 备份和恢复

### 备份数据库

```
pg_dump physicians > physicians_backup.sql
```

### 恢复数据库

```
psql physicians < physicians_backup.sql
```



---

## 性能优化建议

1. **索引优化**: 为经常查询的字段创建索引
  2. **分区**: 对大表（如reviews）考虑按时间分区
  3. **连接池**: 使用连接池管理数据库连接
  4. **查询优化**: 使用EXPLAIN分析慢查询
  5. **缓存**: 对热点数据使用Redis缓存
- 

## 数据迁移

当需要升级数据库结构时，请参考[backend/db/](#)目录下的迁移SQL脚本：

- [migration\\_new\\_workflow.sql](#) - 新工作流迁移
  - [rebuild\\_database.sql](#) - 重建数据库
  - [clean\\_database.sql](#) - 清理数据库
- 

## 监控和维护

### 定期维护任务

```
-- 更新表统计信息
ANALYZE;

-- 清理无用数据
VACUUM;

-- 重建索引（如需要）
REINDEX DATABASE physicians;
```

### 监控查询




```
-- 查看当前活动连接
SELECT * FROM pg_stat_activity WHERE datname = 'physicians';

-- 查看表大小
SELECT
    schemaname,
    tablename,
    pg_size_pretty(pg_total_relation_size(schemaname||'.'||tablename))
as size
FROM pg_tables
WHERE schemaname = 'public'
ORDER BY pg_total_relation_size(schemaname||'.'||tablename) DESC;
```



---

## 版本历史

### v1.2.0 (当前版本)

-  添加machine\_annotation\_evaluations表
-  完善索引结构
-  优化查询性能

### v1.1.0

-  添加tasks和human\_annotations表
-  重构model\_annotations表结构

### v1.0.0

-  基础physicians和reviews表结构