



KB 헬스케어 채용 과제

과제 개요 분석

[본문] 고객건강활동 정보를 수집하여 서비스를 제공하는 당사 상황에 맞춰, App to App으로 삼성헬스나 애플건강으로 단말로 전달되어 서버로 수집되는 데이터 저장 처리를 위한 Backend 비즈니스 프로세스를 구현합니다.

→ 데이터 저장 처리 프로세스를 구현해야함.

Step 1 Input 데이터를 분석하여 수집되는 데이터 분석

Step 2 세부 정책 수립 및 시스템 설계

Step 3 데이터베이스 테이블 설계

Step 4 아키텍처 설계

Step 1

Input 데이터 분석

Input 1~4 포맷 비교표

항목	Input 1, 2	Input 3, 4
출처 이름 (<code>source.name</code>)	SamsungHealth	HealthKit
제조사 (<code>source.product.vendor</code>)	Samsung	Apple Inc
모드 (<code>source.mode</code>)	9	10
memo 필드 포함	✗	✓
타임존 포함 여부	✗ (<code>yyyy-MM-dd HH:mm:ss</code>)	✓ (<code>yyyy-MM-ddTHH:mm:ss +0000</code>)
steps 형태	정수 (int)	소수점 문자열

▼ Input 1,2,3,4

```
{
  "recordkey": "7836887b-b12a-440f-af0f-851546",
  "data": {
    "entries": [
      {
        "period": {
          "from": "2024-11-15 00:00:00",
          "to": "2024-11-15 00:10:00"
        },
        "distance": {
          "unit": "km",
          "value": 0.04223
        }
      }
    ]
  }
}
```

```
{
  "recordkey": "7b012e6e-ba2b-49c7-bc2e-473b",
  "data": {
    "memo": "",
    "entries": [
      {
        "steps": "24",
        "period": {
          "to": "2024-11-14T21:30:00+0000",
          "from": "2024-11-14T21:20:00+0000"
        },
        "distance": {
          "value": 0.0192,

```

```

        "calories": {
          "unit": "kcal",
          "value": 2.03
        },
        "steps": 54
      },
      {
        "period": {
          "from": "2024-11-15 00:50:00",
          "to": "2024-11-15 01:00:00"
        },
        "distance": {
          "unit": "km",
          "value": 0.00553
        },
        "calories": {
          "unit": "kcal",
          "value": 0.24
        },
        "steps": 7
      },
      .....
    ],
    "source": {
      "mode": 9,
      "product": {
        "name": "Android",
        "vender": "Samsung"
      },
      "name": "SamsungHealth",
      "type": ""
    }
  },
  "lastUpdate": "2024-12-16 14:40:00 +0000",
  "type": "steps"
}

```

```

        "unit": "km"
      },
      "calories": {
        "value": 0,
        "unit": "kcal"
      }
    },
    {
      "steps": "688.5509846105425",
      "period": {
        "to": "2024-11-14T21:50:00+0000",
        "from": "2024-11-14T21:40:00+0000"
      },
      "distance": {
        "value": 0.550840787688434,
        "unit": "km"
      },
      "calories": {
        "value": 0,
        "unit": "kcal"
      }
    },
    .....
  ],
  "source": {
    "product": {
      "name": "iPhone",
      "vender": "Apple inc."
    },
    "type": "",
    "mode": 10,
    "name": "Health Kit"
  }
},
"type": "steps",
"lastUpdate": "2024-12-15 12:40:00 +0000"
}

```

Step 2

세부 정책 수립 및 시스템 설계

[Timezone 처리]

Samsung Health 와 Apple HealthKit 에서 제공하는 데이터는 구조는 유사하지만, **steps** 의 소수점 표기 여부와 **period** 필드의 타임존 처리 방식에서 차이를 보입니다.

- **Apple HealthKit:** **period.from** 및 **period.to** 에 **+0000** 과 같은 **UTC 타임존이 명시되어** 있습니다.
- **Samsung Health:** 동일 필드에 **타임존 정보가 포함되어 있지 않으며**, 기기 현지 시간 기준으로 기록된 것으로 판단됩니다.

과제 문서에는 이와 관련된 타임존 처리 기준이 명시되어 있지 않기 때문에, 본 프로젝트에서는 다음과 같은 **가상의 정책**을 수립하고 이를 기반으로 구현을 진행하였습니다:

- **Samsung Health, Apple HealthKit** 데이터 모두 **UTC** 기준으로 처리
 - 따라서 사용자의 시간대 정보에 따라 hours, daily, monthly 통계 결과가 달라질 수 있음.
 - 아래는 기기 시간대를 변경하여 테스트한 결과로, **Apple HealthKit이 수집한 데이터의 시간 처리 방식**을 확인하기 위한 실험입니다. 기기 시간대에 따라 결과의 차이를 확인할 수 있습니다.
- **해결책** : hours 데이터만 따로 적재하여 사용자의 시간대에 따라 daily, monthly 통계에 대한 조희.



KST (UTC+9) 2025년 2월



UTC+14 2025년 2월

[Steps 값 소수점 처리]

수집되는 **steps** 값은 일부 기기에서 **정수**가 아닌 **소수점 단위로 수집**됩니다.

예를 들어, Apple HealthKit은 **0.5 걸음**과 같은 단위를 허용하여, 사용자의 동작을 더 정밀하게 추적합니다.

- 이러한 특성을 반영하여 **steps** 필드는 **DECIMAL(10, 5)** 형식으로 DB에 저장되며, 집계 로직에서도 **소수점 오차 없이 누적 합산**될 수 있도록 **BigDecimal** 을 사용해 처리하였습니다.
- 실 사용 시에는 **소수점 걸음 수** 가 UX적으로 어색할 수 있으므로, 표시는 **반올림** , 통계/누적은 소수점 단위 그대로 처리하는 것으로 가정하였습니다.

Step 3

데이터베이스 테이블 설계

▼ DDL

```
create table if not exists health_db.activity_entry
(
    record_id    bigint                not null comment 'activity_record.id (FK 논리적 참조 - 파티셔닝 제약으로 F
    start_time   datetime              not null comment 'period.from',
    end_time     datetime              not null comment 'period.to',
    steps        decimal(10, 5)        null,
    distance_km  decimal(10, 7)        null,
    calories_kcal decimal(10, 4)        null comment '칼로리 (단위: kcal) - 값의 크기는 작지만 누적합산에 사용
    created_at   datetime default CURRENT_TIMESTAMP null,
    primary key (record_id, start_time)
)
partition by range (to_days(`start_time`)) (
    partition p202410 values less than (739556),
    partition p202411 values less than (739586),
    partition p202412 values less than (739617),
    partition pmax values less than (MAXVALUE)
);

create index idx_entry_record_time
on health_db.activity_entry (record_id, start_time);

create table if not exists health_db.activity_hourly_summary
(
    record_key varchar(64)            not null comment 'record_key',
    hourly     datetime              not null comment '집계 기준 시간 (예: 2024-11-15 08:00)',
    steps       double default 0       null comment '걸음 수',
    calories    double default 0       null comment '소모 칼로리',
    distance    double default 0       null comment '이동 거리 (km)',
    created_at  datetime default CURRENT_TIMESTAMP null,
    updated_at  datetime default CURRENT_TIMESTAMP null on update CURRENT_TIMESTAMP,
```

```

primary key (record_key, hourly)
)
partition by range columns ('hourly') (
    partition p202403 values less than ('2024-11-01'),
    partition p202404 values less than ('2024-12-01'),
    partition p202405 values less than ('2025-01-01'),
    partition pmax values less than (MAXVALUE)
);

create table if not exists health_db.activity_record_source
(
    id          bigint auto_increment
        primary key,
    name        varchar(24) null,
    mode        int         null comment '예: 9 (Samsung), 10 (Apple)',
    type        varchar(12) null comment '예: activity 종류 등',
    product_name varchar(24) null,
    product_vendor varchar(24) null comment '예: Samsung, Apple Inc',
    created_at   datetime(6) null comment '생성 시간',
    constraint UKdic1jjmh40k0njc507m22i0a
        unique (name, mode, type, product_name, product_vendor),
    constraint uk_source_fields
        unique (name, mode, type, product_name, product_vendor)
)
collate = utf8mb4_general_ci;

create table if not exists health_db.activity_record
(
    id          bigint auto_increment
        primary key,
    record_key   varchar(64)          not null comment '중복 방지용 고유 키',
    source_id    bigint              not null comment 'activity_source.id (FK)',
    activity_type enum ('STEPS')      null,
    last_updated_at datetime         null comment '단말 기준 마지막 업데이트 시간 (lastUpdate)',
    created_at   datetime default CURRENT_TIMESTAMP null comment '서버 수신 시간',
    memo         varchar(32)         null comment 'Apple Inc의 HealthKit만 제공 중',
    constraint record_key
        unique (record_key),
    constraint fk_activity_source
        foreign key (source_id) references health_db.activity_record_source (id)
);

```

▼ 집계 SQL

```

# Daily 집계
SELECT DATE(e.start_time)      AS Daily,
       ROUND(SUM(e.steps), 5)  AS Steps,
       ROUND(SUM(e.calories_kcal), 2) AS calories,
       ROUND(SUM(e.distance_km), 2) AS distance,
       r.record_key            AS recordkey
FROM activity_entry e
     JOIN activity_record r ON e.record_id = r.id
WHERE r.record_key = 'e27ba7ef-8bb2-424c-af1d-877e826b7487'
GROUP BY Daily, r.record_key

```

```
ORDER BY Daily, r.record_key;
```

```
# Monthly 집계
```

```
SELECT
```

```
    DATE_FORMAT(e.start_time, '%Y-%m') AS month,
```

```
    ROUND(SUM(e.steps), 0) AS steps,
```

```
    ROUND(SUM(e.calories_kcal), 2) AS calories,
```

```
    ROUND(SUM(e.distance_km), 2) AS distance,
```

```
    r.record_key
```

```
FROM activity_entry e
```

```
JOIN activity_record r ON e.record_id = r.id
```

```
WHERE DATE_FORMAT(e.start_time, '%Y-%m') = '2024-11'
```

```
GROUP BY month, r.record_key
```

```
ORDER BY month, r.record_key;
```

Step 4

[README.md](#) 로 대체