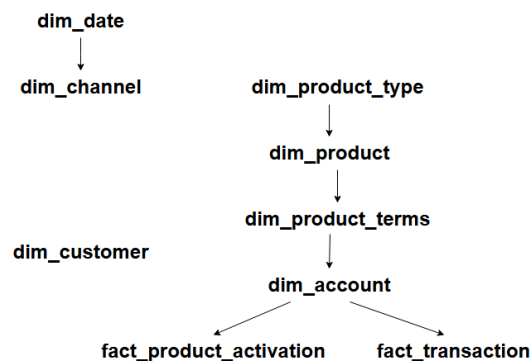


## Задание 6 "ETL и репликация данных"

### 1. Схема потоков перемещения данных из источников в ХД

	Поток	Источник		Приемник	Назначение
1	Генерация календаря	—		dim_date	Поддержка агрегаций по дням, месяцам, кварталам
2	Измерение "Клиент"	client	→	dim_customer	Аналитика по клиентам
3	Измерение "Канал"	channel	→	dim_channel	Анализ по каналам подключения
4	Измерение "Тип продукта"	product.product_type	→	dim_product_type	Группировка продуктов
5	Измерение "Продукт"	product	→	dim_product	Справочник продуктов
6	Измерение "Условия продукта"	product_audit	→	dim_product_terms	SCD Type 2: отслеживание ставок и лимитов
7	Факт "Подключение продукта"	product_application	→	fact_product_activation	Кто и когда подключил продукт
8	Измерение "Счёт"	account	→	dim_account	Привязка счёта к продукту и клиенту
9	Факт "Финансовые операции"	transaction	→	fact_transaction	Движение средств по счетам

### 2. Порядок загрузки данных в таблицы ХД



1. `dim_date` генерируется один раз (например, на 10 лет вперёд), используется всеми фактами: `fact_product_activation`, `fact_transaction`. Без неё невозможно преобразовать `activation_dttm` или `transaction_dttm` в `date_id`

2. `dim_channel` – простой справочник, нет зависимостей. Используется в `fact_product_activation`

3. `dim_product_type` содержит категории: "Кредит", "Вклад", "Карта". Нужен для `dim_product.product_type_id`

Создаётся через `DISTINCT product_type FROM product`

4. `dim_product` зависит от `dim_product_type`, содержит `product_id`, `product_name`, `launch_date`, `status`

Нужен для всех связанных сущностей: `dim_product_terms`, `dim_account`, `fact_product_activation`

5. `dim_product_terms`

SCD Type 2: хранит историю изменений (ставка, лимиты). Зависит от `dim_product`.

6. `dim_customer` – независимое измерение, можно грузить и параллельно с `dim_channel`, но для упрощения ETL можно последовательно

7. `dim_account` зависит от: `dim_product` (через `product_id`) и `dim_customer` (через `client_id`)

Связывает продукт и клиента со счётом. Нужна для `fact_transaction`

8. `fact_product_activation` – факт подключения продукта, зависит от: `dim_customer`, `dim_product`, `dim_channel`, `dim_product_terms` (условия на момент подключения), `dim_date` (по `activation_dttm`).

Все зависимости уже загружены → можно грузить

9. `fact_transaction` – факт финансовой операции. Зависит от: `dim_account` (через `account_id`), `dim_date` (по `transaction_dttm`)

Грузится последней, так как зависит от `dim_account`

### 3. Для каждого отдельного потока данных

- Выберите способ извлечения данных и опишите подробности его применения
- Перечислите уместные преобразования данных
- Подробно опишите S2T и все задействованные сущности

## 1. dim\_date

**Назначение:** справочник дат для агрегаций по времени

**Источник:** генерируется ETL

**Способ извлечения:** генерация данных (Synthetic Generation)

Создаётся диапазон дат (например, с 2020-01-01 по 2030-12-31)

### Преобразования:

- `full_date = DATE '2020-01-01' + n`
- `date_id = TO_CHAR(full_date, 'YYYYMMDD')`
- `day = EXTRACT(day FROM full_date)`
- `month = EXTRACT(month FROM full_date)`
- `year = EXTRACT(year FROM full_date)`
- `quarter = CEIL(EXTRACT(month FROM full_date) / 3)`
- `month_name = TO_CHAR(full_date, 'Month')`
- `is_weekend = EXTRACT(dow FROM full_date) IN (0,6)`

### S2T-маппинг:

Источник	Поле	Приемник	Поле	Трансформация
генератор)	full_date	dim_date	full_date	копирование
(генератор)	full_date	dim_date	date_id	<code>TO_CHAR(full_date, 'YYYYMMDD')</code>
(генератор)	full_date	dim_date	day	<code>EXTRACT(day FROM full_date)</code>
(генератор)	full_date	dim_date	month	<code>EXTRACT(month FROM full_date)</code>
(генератор)	full_date	dim_date	year	<code>EXTRACT(year FROM full_date)</code>
(генератор)	full_date	dim_date	quarter	<code>CEIL(EXTRACT(month FROM full_date)/3)</code>
(генератор)	full_date	dim_date	month_name	<code>TO_CHAR(full_date, 'Month')</code>
(генератор)	full_date	dim_date	is_weekend	<code>EXTRACT(dow FROM full_date) IN (0,6)</code>

## 2. dim\_channel

**Источник:** `channel` из OLTP

**Способ извлечения:** Full Extract (полная выгрузка), так как таблица маленькая и редко меняется.

Преобразования:

- Копирование полей без изменений
- Проверка на дубликаты по `channel_id`
- Очистка от NULL в обязательных полях

S2T-маппинг:

Источник	Поле	Приемник	Поле	Трансформация
channel	channel_id	dim_channel	channel_id	копирование
channel	channel_cd	dim_channel	channel_cd	копирование
channel	channel_name	dim_channel	channel_name	копирование

3. dim\_product\_type

Источник: `product.product_type` из OLTP

Способ извлечения: DISTINCT + Lookup

```
SELECT DISTINCT product_type FROM product WHERE product_type IS NOT NULL
```

Преобразования:

- Удалить пустые/некорректные значения
- Присвоить `product_type_id` (через sequence или auto-increment)
- Нормализовать названия (например, "кредит" → "Кредит")

S2T-маппинг:

Источник	Поле	Приемник	Поле	Трансформация
<code>(SELECT DISTINCT product_type FROM product)</code>	<code>product_type</code>	<code>dim_product_type</code>	<code>product_type_name</code>	нормализация регистра
<code>(SEQUENCE)</code>	—	<code>dim_product_type</code>	<code>product_type_id</code>	присвоить автоинкремент

#### 4. dim\_product

**Источник:** `product` из OLTP

**Способ извлечения:** Incremental Extract по `updated_at` или `created_at`  
Если нет — Full Extract

**Преобразования:**

- `product_type` → заменить на `product_type_id` через lookup к `dim_product_type`
- `status` → нормализовать ("активен", "закрыт")
- Конвертация `launch_date` в формат DATE

**S2T-маппинг:**

Источник	Поле	Приемник	Поле	Трансформация
<code>product</code>	<code>product_id</code>	<code>dim_product</code>	<code>product_id</code>	копирование
<code>product</code>	<code>product_name</code>	<code>dim_product</code>	<code>product_name</code>	копирование
<code>product</code>	<code>description</code>	<code>dim_product</code>	<code>description</code>	копирование
<code>product</code>	<code>brand</code>	<code>dim_product</code>	<code>brand</code>	копирование
<code>product</code>	<code>category</code>	<code>dim_product</code>	<code>category</code>	копирование
<code>product</code>	<code>launch_date</code>	<code>dim_product</code>	<code>launch_date</code>	копирование
<code>product</code>	<code>status</code>	<code>dim_product</code>	<code>status</code>	нормализация
<code>product</code>	<code>product_type</code>	<code>dim_product</code>	<code>product_type_id</code>	<code>lookup(product_type → product_type_id)</code>

#### 5. dim\_product\_terms (SCD Type 2)

**Источник:** `product_audit` из OLTP

**Способ извлечения:** Incremental Extract по `valid_from_dttm` или `created_at`  
Только новые версии условий

**Преобразования:** SCD Type 2 Logic:

- 1. Для каждого `product_id` найти изменения (`interest_rate`, `min_amount`, `max_amount`)
- 2. Если есть изменения:
  - Закрыть предыдущую запись: `valid_to_dttm = new_valid_from - 1 сек`, `is_current = FALSE`
  - Добавить новую: `is_current = TRUE`, `valid_from_dttm`
- 3. Сохранить историю

**S2T-маппинг:**

Источник	Поле	Приемник	Поле	Трансформация
<code>product_audit</code>	<code>audit_id</code>	<code>dim_product_terms</code>	<code>product_terms_id</code>	копирование или sequence
<code>product_audit</code>	<code>product_id</code>	<code>dim_product_terms</code>	<code>product_id</code>	копирование
<code>product_audit</code>	<code>interest_rate</code>	<code>dim_product_terms</code>	<code>interest_rate</code>	копирование
<code>product_audit</code>	<code>min_amount</code>	<code>dim_product_terms</code>	<code>min_amount</code>	копирование
<code>product_audit</code>	<code>max_amount</code>	<code>dim_product_terms</code>	<code>max_amount</code>	копирование
<code>product_audit</code>	<code>valid_from_dttm</code>	<code>dim_product_terms</code>	<code>valid_from_dttm</code>	копирование
<code>product_audit</code>	<code>valid_to_dttm</code>	<code>dim_product_terms</code>	<code>valid_to_dttm</code>	копирование
<code>product_audit</code>	<code>is_current</code>	<code>dim_product_terms</code>	<code>is_current</code>	копирование (или рассчитывается в ETL)

6. dim\_customer

Источник: client из OLTP

Способ извлечения: Incremental Extract по updated\_at или created\_at

Преобразования:

- Проверка на дубликаты по passport\_number
- Маскировка паспорта (если нужно для безопасности)
- Конкатенация ФИО (опционально)

S2T-маппинг:

Источник	Поле	Приемник	Поле	Трансформация
client	client_id	dim_customer	customer_id	копирование
client	first_name	dim_customer	first_name	копирование
client	last_name	dim_customer	last_name	копирование
client	birth_date	dim_customer	birth_date	копирование
client	passport_series	dim_customer	passport_series	копирование
client	passport_number	dim_customer	passport_number	копирование

## 7. `dim_account`

**Источник:** `account` из OLTP

**Способ извлечения:** Incremental Extract по `open_date` или `created_at`

**Преобразования:**

- `currency` → нормализовать (RUB → Рубль)
- `status` → определить: открыт/закрыт (по `close_date`)
- Связь с `customer_id` и `product_id`

**S2T-маппинг:**

Источник	Поле	Приемник	Поле	Трансформация
<code>account</code>	<code>account_id</code>	<code>dim_account</code>	<code>account_id</code>	копирование
<code>account</code>	<code>account_number</code>	<code>dim_account</code>	<code>account_number</code>	копирование
<code>account</code>	<code>currency</code>	<code>dim_account</code>	<code>currency</code>	нормализация
<code>account</code>	<code>open_date</code>	<code>dim_account</code>	<code>open_date</code>	копирование
<code>account</code>	<code>close_date</code>	<code>dim_account</code>	<code>close_date</code>	копирование
<code>account</code>	<code>product_id</code>	<code>dim_account</code>	<code>product_id</code>	копирование
<code>account</code>	<code>client_id</code>	<code>dim_account</code>	<code>customer_id</code>	копирование



8. fact\_product\_activation

Источник: product\_application из OLTP

Способ извлечения: Incremental Extract по activation\_dttm

Преобразования:

- activation\_dttm → activation\_date\_id = TO\_CHAR(activation\_dttm, 'YYYYMMDD')
- Поиск product\_terms\_id:

```
SELECT product_terms_id FROM dim_product_terms

WHERE product_id = f.product_id AND activation_dttm BETWEEN
valid_from_dttm AND COALESCE(valid_to_dttm, NOW())
```

S2T-маппинг:

Источник	Поле	Приемник	Поле	Трансформация
product_application	activation_id	fact_product_activation	activation_id	копирование
product_application	client_id	fact_product_activation	customer_id	копирование
product_application	product_id	fact_product_activation	product_id	копирование
product_application	channel_id	fact_product_activation	channel_id	копирование
product_application	activation_dttm	fact_product_activation	activation_date_id	TO_CHAR(activation_dttm, 'YYYYMMDD')
(lookup)	—	fact_product_activation	product_terms_id	поиск по условиям на момент активации
(константа)	—	fact_product_activation	quantity	1

product_application	status	fact_product_activation	status	копирование
---------------------	--------	-------------------------	--------	-------------

9. fact\_transaction

Источник: transaction из OLTP

Способ извлечения: Incremental Extract по transaction\_dttm

Преобразования:

- transaction\_dttm → transaction\_date\_id = TO\_CHAR(transaction\_dttm, 'YYYYMMDD')
- amount → сохранить как есть
- transaction\_type → МОЖНО ВЫНЕСТИ В dim\_transaction\_type

S2T-маппинг:

Источник	Поле	Приемник	Поле	Трансформация
transaction	transaction_id	fact_transaction	transaction_id	копирование
transaction	account_id	fact_transaction	account_id	копирование
transaction	amount	fact_transaction	amount	копирование
transaction	transaction_dttm	fact_transaction	transaction_date_id	TO_CHAR(transaction_dttm, 'YYYYMMDD')
transaction	transaction_type	fact_transaction	transaction_type	копирование или lookup

