UML Original Schema (Diagram)

Transaction (operational)
- Id
- Source account
- Destination account
- Sum
- Date and time

Account
- Id
- Account Owner
- System account
- Balance

Status
- Id
- Name

Office
- Id
- City
- Name
- Total sells sum

Account Owner
- Id
- Name

Transaction (archive)
- Id
- Source account
- Destination account
- Sum
- Date and time

Site Page
- Id
- Parent page
- Name

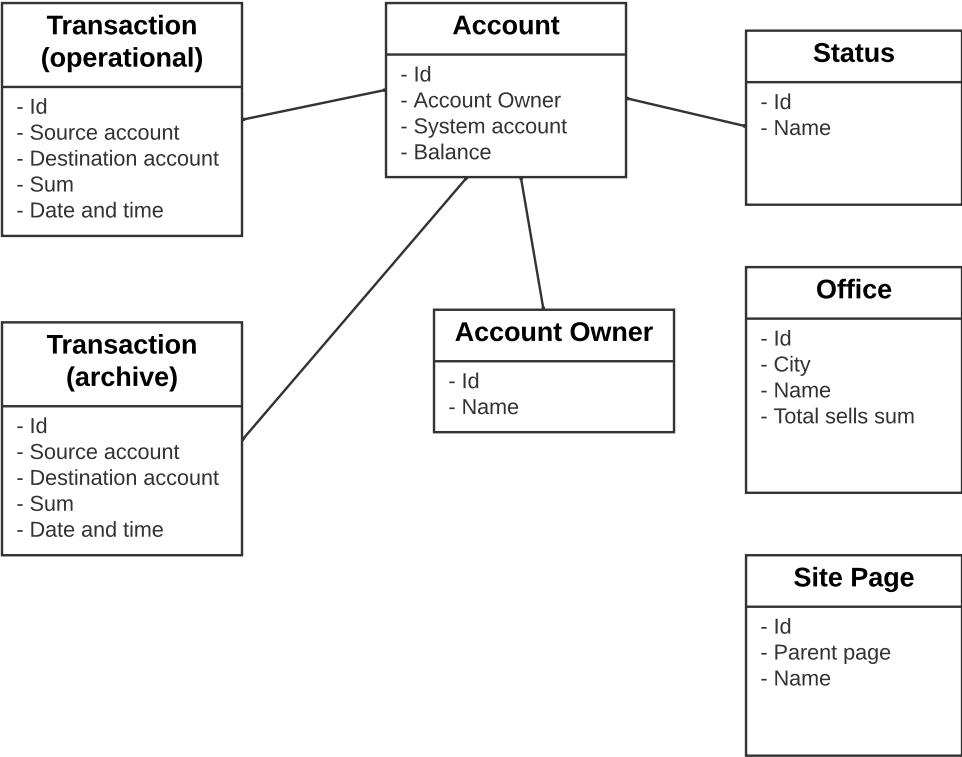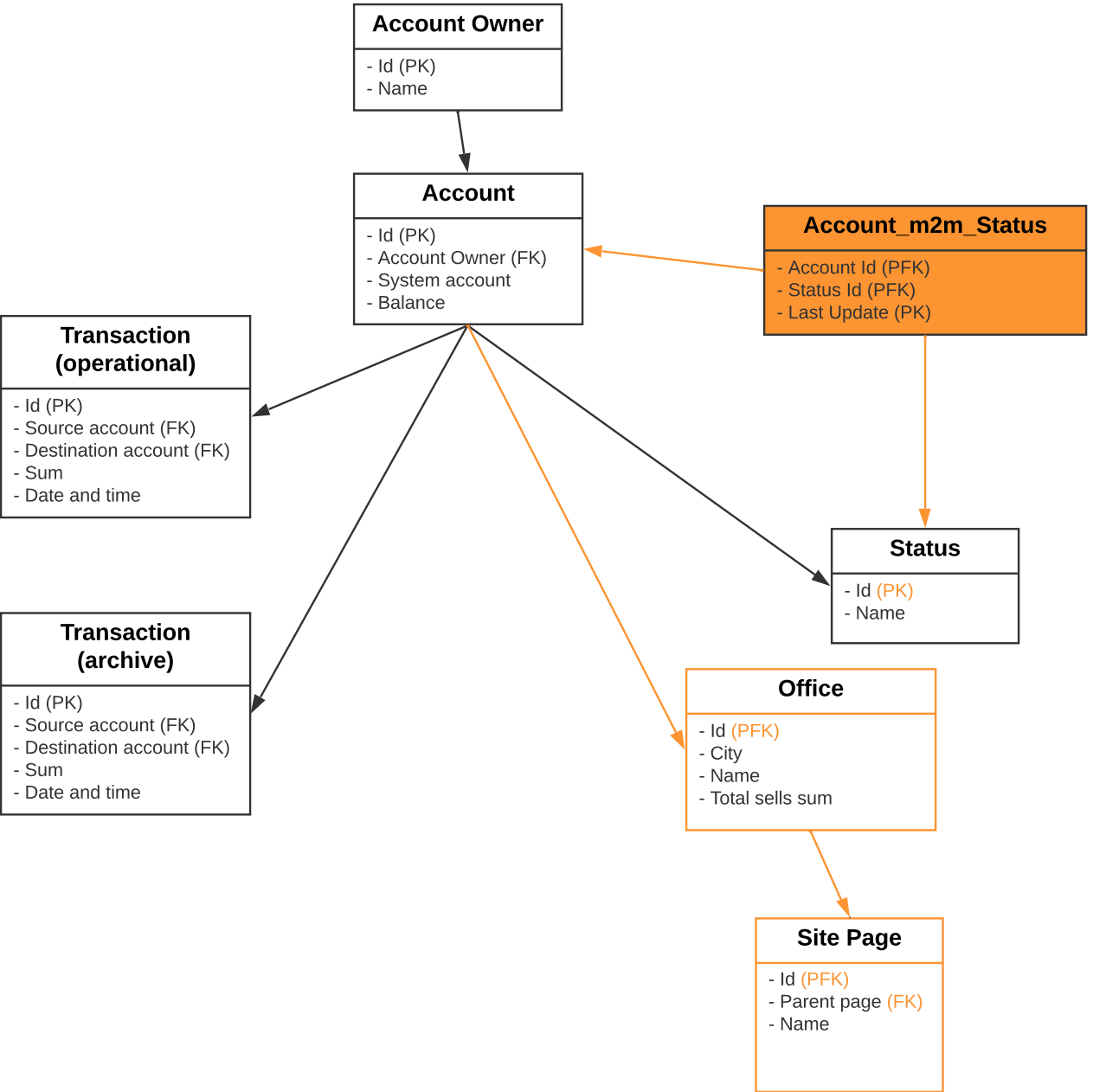
1. What attributes are missing from existing relations? Add those missing attributes.
2. What relations are missing from the database? Add those relations.

(Second Diagram)

Account Owner
- Id (PK)
- Name

Account
- Id (PK)
- Account Owner (FK)
- System account
- Balance

Account_m2m_Status
- Account Id (PFK)
- Status Id (PFK)
- Last Update (PK)

Transaction (operational)
- Id (PK)
- Source account (FK)
- Destination account (FK)
- Sum
- Date and time

Status
- Id (PK)
- Name

Transaction (archive)
- Id (PK)
- Source account (FK)
- Destination account (FK)
- Sum
- Date and time

Office
- Id (PFK)
- City
- Name
- Total sells sum

Site Page
- Id (PFK)
- Parent page (FK)
- Name

---

7. What cascade operations are necessary in this database? Make a list.

ON DELETE:
- of_id --> sp_id // optional
  - if an office city is removed, so should the be the corresponding webpage
- a_id --> to_source
- a_id --> to_destination
- a_id --> ta_source
- a_id --> ta_destination
- a_id --> of_id // optional
- sp_parent --> sp_id

ON UPDATE:
- a_id --> to_source
- a_id --> to_destination
- a_id --> ta_source
- a_id --> ta_destination
- o_id --> a_id // optional
- sp_id --> sp_parent

ON DELETE SET NULL:

ON DELETE SET DEFAULT:
- a_id --> ams_a_id --> ams_s_id --> s_id

ON DELETE NO ACTION:
- a_id --> o_id

ON UPDATE NO ACTION:
- sp_parent --> sp_id

(Requirements text)

- **Account** (describes an account):
  - o id (account id);
  - o balance (account balance, MONEY data type);
  - o account owner (FK);
  - o system account (a flag representing that this account does not belong to a human)
- **Status** (account status, e.g., «Active», «Locked» и т.д.):
  - o id (status id);
  - o name (status name).
- **Transaction operational** (for transactions in the current month):
  - o id (transaction id);
  - o source account (FK);
  - o destination account (FK);
  - o date and time (transaction datetime);
  - o sum (transaction sum).
- **Transaction archive** (for transactions before the current month):
  - o id (transaction id);
  - o source account (FK);
  - o destination account (FK);
  - o date and time (transaction datetime);
  - o sum (transaction total sum).
- **Account owner** (bank client):
  - o id (account owner id);
  - o name (account owner name).
- **Site page** (bank website page):
  - o id (page id);
  - o parent page (rFK);
  - o name (page name).
- **Office** (bank office):
  - o id (office id);
  - o city (office location);
  - o name (office name);
  - o total sells sum (MONEY datatype, sum of profit by the office).

Analyze the "Bank" database model (see below). Perform the following tasks:
1. What attributes are missing from existing relations? Add those missing attributes.

2. What relations are missing from the database? Add those relations.
- Acount m2m status
- Account --> office
- Office -> pages

3. Do all objects in the database follow unified naming conventions? Correct those violating the convention.
4. Do all attributes in the database have optimal data types? Propose optimization.
5. Do all relations in the database have optimal keys? If no, propose optimization.
6. Are there redundant or missing relationships in the database? Make necessary corrections.
7. What cascade operations are necessary in this database? Make a list.
8. Can we increase performance using caching relations? If yes, add such relations.
9. What indexes are missing from the database? Add all necessary indexes.
10. Update logical models in "CTECH DB 02 - Relations, Keys, Relationships, Indexes - task.eap" file, export your final models as SQL scripts and import in a real DBMSes (MS SQL Server and Oracle).