

# Databasmodellering

## Exam

2024-03-21

av

Abduvohid Ilhomov  
abil23@student.bth.se

## Innehåll

Introduktion .....	3
Konceptuell modellering () .....	3
Beskriv databasen i ett textstycke.....	3
Skriv ned alla entiteter .....	3
Skriv ned alla relationer och visa i matris .....	4
Rita enkelt ER-diagram med entiteter och relationer.....	5
Komplettera ER-diagram med kardinalitet .....	6
Komplettera ER-diagram med alla attribut samt kandidatnycklar .....	6
Logisk modellering ().....	6
Modifiera ER-diagram enligt relationsmodellen.....	6
Utöka ER-diagram med primära/främmande nycklar samt kompletterande attribut.....	6
Fysisk modellering () .....	7
Skapa SQL DDL för tabellerna .....	7
Lista funktioner som databasen skall stödja (API) .....	7
APPENDIX DDL.....	8
REFERENSER .....	10

# Introduktion

En databasmodell av exam enligt kokboken [1].

## Konceptuell modellering

### Beskriv databasen i ett textstycke

1. **Tabellen product** ger en översikt av viktiga AI-produkter som har funnits genom historien. Den innehåller detaljer såsom produktnamn, smeknamn, debattämnen kopplade till produkten, utgivningsår, och en koppling till utvecklaren via ett `developer_id`. Denna tabell är central för att förstå de enskilda AI-produkternas bidrag och historiska kontext.
2. **Tabellen type** beskriver olika kategorier eller typer av AI-teknologi. Varje typ har ett unikt id och ett namn som beskriver teknologikategorin, exempelvis "Stored-program computer" eller "Chess-playing expert system".
3. **Tabellen product2type** är en kopplingstabell som skapar en många-till-många-relation mellan produkter och typer. Detta är nödvändigt eftersom en produkt kan klassificeras under flera typer. Tabellen lagrar `product_id` och `type_id` för att koppla samman de relevanta posterna i product- och type-tabellerna.
4. **Tabellen developer** innehåller detaljer om de individer, grupper eller organisationer som har utvecklat de olika AI-produkterna. Den inkluderar information såsom utvecklarens namn och land.

### Skriv ned alla entiteter

[product] En översikt av viktiga AI-produkter som funnits i historien.

[type] Detaljer om typer, produkter kan vara av olika typer.

[product2type] Kopplingstabell mellan "product" och "type".

[developer] Detaljer om vem som utvecklat en viss produkt.

Skriv ned alla relationer och visa i matris

	Product	Type	Developer	Product2Type
Product			1	
Type				
Developer				
Product2Type	M	M		

Rita enkelt ER-diagram med entiteter och relationer

### 1. **Product**

- id (PK)
- name
- nick
- debate
- year
- developer\_id (FK)

### 2. **Type**

- id (PK)
- name

### 3. **Developer**

- id (PK)
- developer
- country

### 4. **Product2Type**

- product\_id (FK)
- type\_id (FK)

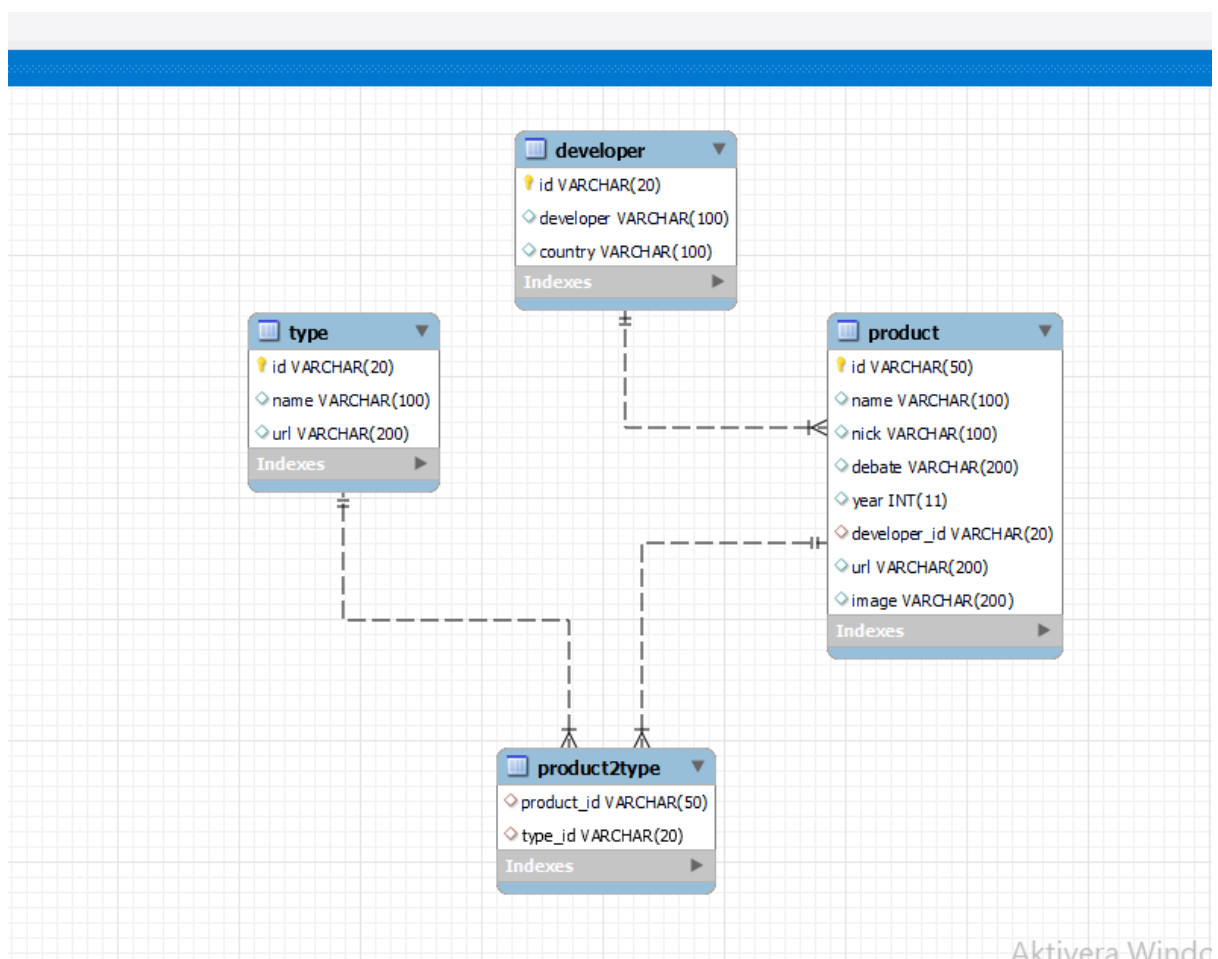
Komplettera ER-diagram med kardinalitet

Komplettera ER-diagram med alla attribut samt kandidatnycklar

## Logisk modellering

Modifiera ER-diagram enligt relationsmodellen

Utöka ER-diagram med primära/främmande nycklar samt kompletterande attribut



# Fysisk modellering ()

Skapa SQL DDL för tabellerna

Lista funktioner som databasen skall stödja (API)

Man ska kunna visa tabellerna med och utan url, man ska kunna söka för en specifik kolumn så tabellen blir annorlunda.

# APPENDIX DDL

```
CREATE DATABASE IF NOT EXISTS exam;

use exam;

DROP TABLE IF EXISTS product2type;
DROP TABLE IF EXISTS product;
DROP TABLE IF EXISTS developer;
DROP TABLE IF EXISTS type;

CREATE TABLE type(
    id VARCHAR(20) PRIMARY KEY,
    name VARCHAR(100),
    url VARCHAR(200)
);

CREATE TABLE developer(
    id VARCHAR(20) PRIMARY KEY,
    developer VARCHAR(100),
    country VARCHAR(100)
);

CREATE TABLE product(
    id VARCHAR(50) PRIMARY KEY,
    name VARCHAR(100),
    nick VARCHAR(100),
    debate VARCHAR(200),
    year INT,
    developer_id VARCHAR(20),
    url VARCHAR(200),
    image VARCHAR(200),
    FOREIGN KEY (developer_id) REFERENCES developer(id)
);

CREATE TABLE product2type(
    product_id VARCHAR(50),
    type_id VARCHAR(20),
    FOREIGN KEY (product_id) REFERENCES product(id),
    FOREIGN KEY (type_id) REFERENCES type(id)
);
```



```
DROP PROCEDURE IF EXISTS show_all_products;
DELIMITER ;;
CREATE PROCEDURE show_all_products()
BEGIN
    SELECT
        p.id,
        p.name AS pname,
        p.nick,
        p.debate,
        p.year,
        p.developer_id,
        t.id,
        t.name,
        pt.product_id,
        pt.type_id,
        p.url AS purl,
        t.url
    FROM product p
    LEFT JOIN product2type pt ON p.id = pt.product_id
    LEFT JOIN type t ON pt.type_id = t.id
    ORDER BY p.id;
END;;
DELIMITER ;

DROP PROCEDURE IF EXISTS show_all_products_without_url;
DELIMITER ;;
CREATE PROCEDURE show_all_products_without_url()
BEGIN
    SELECT
        p.id,
        p.name AS pname,
        p.nick,
        LEFT(p.debate, 20) AS debate,
        p.year,
        p.developer_id,
        t.id,
        t.name,
        pt.product_id,
        pt.type_id
    FROM product p
    LEFT JOIN product2type pt ON p.id = pt.product_id
    LEFT JOIN type t ON pt.type_id = t.id
    ORDER BY p.id;
END;;
DELIMITER ;

DROP PROCEDURE IF EXISTS search_all_products;
```

```

DELIMITER ;;
CREATE PROCEDURE search_all_products(search VARCHAR(100))
BEGIN
    SELECT
        p.id,
        p.name AS pname,
        p.nick,
        p.debate,
        p.year,
        p.developer_id,
        t.id,
        t.name,
        pt.product_id,
        pt.type_id,
        p.url AS purl,
        t.url
    FROM product p
    LEFT JOIN product2type pt ON p.id = pt.product_id
    LEFT JOIN type t ON pt.type_id = t.id
    WHERE CONCAT(p.id, p.name, p.nick, p.debate, p.year, p.developer_id, t.id,
        t.name, pt.product_id, pt.type_id) LIKE search
    ORDER BY p.id;
END;;
DELIMITER ;

```

## REFERENSER

[1] Kokbok för databasmodellering, <https://dbwebb.se/kunskap/kokbok-for-databasmodellering>, visited 2023-01-31.