

Application description

Projekt byl založen na databázi knihkupectví. Pomocí této databáze můžete řešit základní dotazy, které budou v rámci obchodu vyžadovány. například třídění knih podle jejich autorů nebo vydavatelů nebo sledování objednávek (adresa, čas a datum jejich příchodu; počet knih v každé objednávce az toho kalkulovaná cena celé objednávky, určení identity zakazníka atd.). Funkčnost také zahrnuje různé akce se všemi stranami interagujícími s obchodem (zaměstnanec, autor, kupující) a jejich další atributy, jako jsou adresy, pracovní pozice, kontaktní údaje, objednávky jednotlivců, slevy.

Příklady některých use-case:

- Vyhledování názvu knihy podle vydavatele:

```
select title from book b join book_has_publisher p on b.book_id = p.book_id join publisher on  
p.publisher_id=publisher.publisher_id where publisher = 'Penguin Random House';
```

- Vyhledování jmen a plátů všech pracovníků v databázi:

```
select first_name, last_name, salary from person p join employee e on p.employee_id = e.employee_id  
where p.employee_id is not null;
```

- Vyhledování objednaných knih a id kupujícího

```
select "title", "customer_id" from "book" b join "order_has_book" o on b.book_id = o.book_id join  
"order" on "order".order_id = o.order_id;
```

Description of each table

- Table „person“: v tabulce person ukládají osobní údaje jednotlivců (person_id SERIAL Primary key, jmeno varchar(45), příjmení varchar(45), password varchar(45), email varchar(45)) a taky údaje o roli jednotlivců vůči obcodu, prostředctvím FK (employee_id INT, author_id INT, customer_id INT) .
- Tables „employee“, „customer“, „author“ : vztahují se k tabulce person a taky mají svoji vlastní atributy(salary FLOAT, login varchar(45), pseudonym(45)).
- Tables „position“, „disccount“ : vztahují se k tabulkám „empoloyee“ a „ customer“ jako dopolňující tabulky pro určení jedné z více pozice (position varchar(45)) nebo určení konkrétního typu slevy(disccount_type varchar(45), discount_quantity INT).
- Table „contact“ : slouží pro případné určení telefonního čísla osoby(phone_number INT).
- Table „address“: slouží pro určení adresy osoby nebo pro určení adresy (city varchar(45), district varchar(45), street varchar(45), numer varchar(45), post_code varchar(45)) dodávky objednávky např. na pobočku.
- Table „genre“ : slouží pro vypis všech žánrů v databázi (genre varchar(45)).
- Table „publisher“ : slouží pro vypis všech vydavatelů v databázi (publisher varchar(45))
- Table “book” : obsahuje všechna podstatná data o knichech (book_id SERIAL Pimary key, title varchar(45), year_of_issue INT, price FLOAT) taky je v relacích s „publisher“, „author“, „genre“, „order“ prostředctvím FK.
- Table „order“: obsahuje všechna podstatná data o objednavkách (order_id SERIAL Primary key, date DATE, time TIME, count_of_books INT) a taky je propojená s “address”, “customer”, “book”.
- Tables „order_has_book“, „person_has_address“, „book_has_author“, „bool_has_genre“, book_has_publisher“: pro jednoznačné určení vztáhu v relacích typu N ku M.

DATA TYPE: SERIAL se používal pro jednoduché generování primárních klíčů;

DATA TYPE: varchar se používal pro popis atributů pomocí String;

DATA TYPE: time, date se používali pro popis daty a času v podstaném formátu;

DATA TYPE: INT se používal při jednoduchém popisu celá čísla

why do you think that your database is in 3rd normal form

- **First Normal Form:** A relational schema R is in first normal form if the domains of all attributes of R are atomic; což platí pro mou databázi, jelikož všechny atributy v tabulkách jsou uvedeny atomické. Příklad: person_id SERIAL Primary key, jmeno varchar(45), příjmení varchar(45), password varchar(45), email varchar(45))
- **Second Normal Form:** the database is in the 1NF ; Every non-key attribute of relation R is fully dependent on every candidate key of relation R; což platí, jelikož tabulky mám v 1NF a v každé tabulce jsou jenom atributy závislé na PK.
- **Third Normal Form:** the database is in the 2NF All transitive dependencies must be removed; a non-key attribute may not be functionally dependent on another non-key attribute; což platí jelikož tabulky jsou v 2NF a jsou odstraněny všechny tranzitivní závislosti. Příklad: tabulka "book" a tabulka "author" odstraněna tranzitivní závislost jména autora na názvu knihy.