

# **Politechnika Krakowska**

## **im. Tadeusza Kościuszki**

### **Wydział Inżynierii Elektrycznej i Komputerowej**

**Zaawansowane Bazy Danych**  
**Prowadzący: mgr inż. Krzysztof Czajkowski**

**Wykonali:** Wojciech Juszczak, Cezary Śląwecki  
**Grupa:** 43i

**Kierunek:** Informatyka  
**Rok:** IV  
**Semestr:** 7

**Projekt – Hurtownia danych**

# **Spis treści**

## **1. Wprowadzenie**

## **2. Ewolucja projektu**

1. Stworzenie diagramów oraz kodu SQL baz
2. Pierwsza wersja hurtowni
3. *Table czasu*
4. Poprawki do *Staging Area*; mapowanie użytkowników
5. Mapowanie kategorii
6. Kolejne aktualizacje do *Staging Area* oraz baz
7. Widok *Kategorie* oraz poprawki do pozostałych widoków
8. Widok dla *Tabeli faktów*
9. Aktualizacja tabeli *Czas*
10. Partycjonowanie *Tabeli faktów*

## **3. Widoki**

## **4. Zapytania**

# 1. Wprowadzenie

Dokumentacja przedstawia proces tworzenia hurtowni danych, baz danych, opis ich przekształcania ku ostatecznej formie.

Nasz projekt zawiera dwie bazy danych, każda stworzona przez jednego członka zespołu. Bazy te zawierają dane będące umownym odzwierciedleniem rzeczywistych informacji gromadzonych przez internetowe serwisy aukcyjne.

# 2. Ewolucja projektu

1. Projekt zaczęliśmy od stworzenia diagramów ERD naszych baz tak, aby zawierały one ~10 encji. Na podstawie utworzonych diagramów napisaliśmy kod SQL generujący zamierzone bazy danych.

Poniżej znajduje się fragment prezentujący tworzenie jednej z tabel, dodawanie sekwencji oraz wypełnianie tej tabeli.

```
CREATE TABLE aukcja
(
    id number(10) PRIMARY KEY,
    nazwa varchar2(32) NOT NULL,
    typ number(1) NOT NULL,
    opis varchar(128) NOT NULL,
    cenaMinimalna number(12,2) NOT NULL,
    cena number(12,2) NOT NULL,
    dataUtworzenia Timestamp NOT NULL,
    dataZakonczenia Timestamp NOT NULL,
    idKomentarza number(10) NOT NULL,
    idPrzedmiotu number(10) NOT NULL
);
-- idKomentarza number(10) FOREIGN KEY,
-- idPrzedmiotu number(10) FOREIGN KEY

ALTER TABLE aukcja ADD CONSTRAINT fk_komentarz1 FOREIGN KEY (idKomentarza)
REFERENCES komentarz(id);
ALTER TABLE aukcja ADD CONSTRAINT fk_przedmiot FOREIGN KEY (idPrzedmiotu)
REFERENCES przedmiot(id);

CREATE SEQUENCE aukcja_seq START WITH 0 INCREMENT BY 1 MINVALUE 0 NOCACHE
NOCYCLE;

-- Generacja 100000 aukcji
insert into aukcja
    (id, nazwa, typ, opis, cenaMinimalna, cena, dataUtworzenia,
    dataZakonczenia, idKomentarza, idPrzedmiotu)
    select aukcja_seq.NEXTVAL,
DBMS_RANDOM.string('X',TRUNC(DBMS_RANDOM.value(1,32))),
DBMS_RANDOM.value(0,1),
DBMS_RANDOM.string('X',TRUNC(DBMS_RANDOM.value(1,128))),
DBMS_RANDOM.value(0,9999.99), DBMS_RANDOM.value(0,9999.99), TO_TIMESTAMP
('2000/01/01 00:00:00.000000', 'YYYY/MM/DD'
HH24:MI:SS.FF')+numtodsinterval(DBMS_RANDOM.value()* 259200,'SECOND')
+NUMTOYMINTERVAL(DBMS_RANDOM.value(0,240), 'month'), TO_TIMESTAMP
```

```

('2000/01/01 00:00:00.000000', 'YYYY/MM/DD'
HH24:MI:SS.FF')+numtodsinterval(DBMS_RANDOM.value()* 259200,'SECOND')
+NUMTOYMINTERVAL(DBMS_RANDOM.value(0,240),'month'),
DBMS_RANDOM.value(0,99999), DBMS_RANDOM.value(0,99999)
from dual
connect by level <= 100000;

```

2. Za następny etap uznaliśmy stworzenie hurtowni, co osiągnęliśmy wybierając interesujące nas kolumny z tabel tj. te, z których można uzyskać informacje do ostatecznych zapytań dla hurtowni.

Oto kod analogicznie tworzący tabele dla o aukcjach dla hurtowni, bez niepotrzebnych kolumn:

```

CREATE TABLE oferty
(
    id number(10) PRIMARY KEY,
    dataUtworzenia Timestamp NULL,
    dataZamkniecia Timestamp NULL,
    liczbaWyswietlen number(10) NULL,
    typOferty number(1) NULL,
    cena number(10) NULL,
    cenaMinimalna number(12,2) NULL,
    idLicytacji number(10) NULL
);

CREATE SEQUENCE oferty_seq START WITH 0 INCREMENT BY 1 MINVALUE 0 NOCACHE
NOCYCLE;

INSERT INTO oferty
    (id, dataUtworzenia, dataZamkniecia, liczbaWyswietlen, typOferty, cena,
    cenaMinimalna, idLicytacji)
    SELECT oferty_seq.nextval, PO.Data_utworzenia, PO.Data_zamkniecia,
    PO.Liczba_wyswietlen, PO.Typ_oferty, PO.Cena, null, PO.ID_Licytacji
    FROM "PARAGON"."OFERTA" PO;

INSERT INTO oferty
    (id, dataUtworzenia, dataZamkniecia, liczbaWyswietlen, typOferty, cena,
    cenaMinimalna, idLicytacji)
    SELECT oferty_seq.nextval, IA.dataUtworzenia, IA.dataZakonczenia, null,
    IA.typ, null, IA.cenaMinimalna, null
    FROM "IMMORTAL"."AUKCJA" IA;

```

Jak widać dane brane są z obu baz, zależnie od tego, które pola pokrywają się z tymi w hurtowni.

Po dojściu do tego etapu w projekcie uznaliśmy, że to co stworzyliśmy do tej pory to nie jest jeszcze hurtownia, lecz Staging Area.

**3.** Następnym celem więc było utworzenie tabeli czasu dla hurtowni.

```
CREATE TABLE Czas
(
    czas timestamp PRIMARY KEY,
    dzien number,
    miesiac number,
    kwartal number,
    rok number
);
```

Po utworzeniu tabeli napisaliśmy kod wypełniający ją odpowiednimi wartościami, tzn

- „czas” – zawiera dokładny, pełny czas
- „dzien”, „miesiac”, „kwartal”, „rok” – zawierają numer odpowiadającego im dnia, miesiąca, kwartału, roku

```
Declare
Obecny_czas Timestamp;
Obecny_godzina Number;
Obecny_pora_dnia Number;
Obecny_dzien Number;
Obecny_miesiac Number;
Obecny_kwartal Number;
Obecny_rok Number;
Koncowy_czas Timestamp := TO_TIMESTAMP ('2010/12/31 00:00:00.000000',
'YYYY/MM/DD HH24:MI:SS.FF');
Czas_poczatkowy Timestamp := TO_TIMESTAMP ('2000/01/01 00:00:00.000000',
'YYYY/MM/DD HH24:MI:SS.FF');
Ilosc_Wpisow Integer := extract(DAY FROM (Koncowy_czas -Czas_poczatkowy));

BEGIN
FOR i IN 0 .. Ilosc_Wpisow LOOP
    Obecny_czas:=Czas_poczatkowy+numtodsinterval(i*6,'HOUR');
    Obecny_godzina:=extract(HOUR FROM Obecny_czas);
    Obecny_dzien:=extract(DAY FROM Obecny_czas);
    Obecny_miesiac:=extract(MONTH FROM Obecny_czas);
    Obecny_rok:=extract(YEAR FROM Obecny_czas);
    KWARTAL(Obecny_miesiac,Obecny_Kwartal);
    PORA_DNIA(Obecny_pora_dnia,Obecny_godzina);
    INSERT INTO Czas
    (czas,dzien,miesiac,kwartal,rok)
    VALUES
    (Obecny_czas,Obecny_dzien,Obecny_miesiac,Obecny_Kwartal,Obecny_rok);
END LOOP;
END;
```

Stworzyliśmy także procedurę do wyznaczania pory dnia na podstawie godziny tj. cztery pory dnia, każda trwająca sześć godzin.

```
create or replace PROCEDURE OBLICZ_PORE_DNIA
(
    CZAS IN TIMESTAMP,
    PORA_DNIA OUT TIMESTAMP
) AS

Godzina Number;
Dzien Number;
Miesiac Number;
Rok Number;
BEGIN
    Godzina:=extract(HOUR FROM CZAS);
    Dzien:=extract(DAY FROM CZAS);
    Miesiac:=extract(MONTH FROM CZAS);
    Rok:=extract(YEAR FROM CZAS);
    PORA_DNIA:=TO_TIMESTAMP
    (TO_CHAR(Rok)||'/'||TO_CHAR(Miesiac)||'/'||TO_CHAR(Dzien)||'
    00:00:00.000000', 'YYYY/MM/DD HH24:MI:SS.FF');
    CASE
        when Godzina < 6 then null;
        when Godzina >=6 AND Godzina < 12 then PORA_DNIA:=
PORA_DNIA+numtodsinterval(6,'HOUR');
        when Godzina >=12 AND Godzina < 18 then
PORA_DNIA:=PORA_DNIA+numtodsinterval(12,'HOUR');
        when Godzina >=18 AND Godzina < 24 then
PORA_DNIA:=PORA_DNIA+numtodsinterval(18,'HOUR');
        else dbms_output.put_line('Not month!');
    END CASE;
END OBLICZ_PORE_DNIA;
```

4. Zauważliśmy, że podczas przenoszenia rekordów do Staging Area pojawiały się problemy z niespójnością danych, ponieważ nie były zachowywane relacje między użytkownikami a ofertami. Spowodowane to było nadawaniem nowych ID przy przenoszeniu.

Poradziliśmy sobie z tym problemem tworząc mapowanie na ID użytkownika, aby „przypisać” go do danej bazy.

```
CREATE TABLE staging.mapa_Uzytkownicy
(
    id_staging number(10) ,
    id_base number(10) ,
    source_type number(1)
);

CREATE SEQUENCE staging.mapa_Uzytkownicy_seq START WITH 0 INCREMENT BY 1
MINVALUE 0 NOCACHE NOCYCLE;
```

```
...
```

```
INSERT INTO staging.mapa_Uzytkownicy (id_staging,id_base,source_type)
SELECT staging.mapa_Uzytkownicy_seq.nextval ,IU.ID as id_base ,0 as
source_type FROM "IMMORTAL"."UZYTKOWNIK" IU ;

INSERT INTO staging.mapa_Uzytkownicy (id_staging,id_base,source_type)
SELECT staging.mapa_Uzytkownicy_seq.nextval,PU.ID_UZYTKOWNIKA as id_base ,1
as source_type FROM "PARAGON"."UZYTKOWNIK" PU;
ALTER TABLE faktyHurtownia ADD CONSTRAINT fk_idOferty FOREIGN KEY
(idOferty) REFERENCES Oferty(id);
ALTER TABLE faktyHurtownia ADD CONSTRAINT fk_idOfertyWyPr FOREIGN KEY
(idOfertyWyPr) REFERENCES OfertyWygranePrzegrane(id);
ALTER TABLE faktyHurtownia ADD CONSTRAINT fk_idSpory FOREIGN KEY (idSpory)
REFERENCES Spory(id);
ALTER TABLE faktyHurtownia ADD CONSTRAINT fk_idUzytkownicyHurtownia FOREIGN
KEY (idUzytkownicyHurtownia) REFERENCES UzytkownicyHurtownia(id);
ALTER TABLE faktyHurtownia ADD CONSTRAINT fk_ididWyrOff FOREIGN KEY
(idWyrOff) REFERENCES WyroznienieOferty(id);
ALTER TABLE faktyHurtownia ADD CONSTRAINT fk_czas FOREIGN KEY (czasFakty)
REFERENCES
Czas(czas);
```

5. Na tej samej zasadzie utworzyliśmy także mapowanie dla tabeli *Kategoria*:

```
CREATE TABLE staging.mapa_Kategorie
(
    id_staging number(10) ,
    id_base number(10) ,
    source_type number(1)
);

CREATE SEQUENCE staging.mapa_Kategorie_seq START WITH 0 INCREMENT BY 1
MINVALUE 0 NOCACHE NOCYCLE;

INSERT INTO staging.mapa_Kategorie (id_staging,id_base,source_type)
SELECT staging.mapa_Kategorie_seq.nextval ,IK.id as id_base ,0 as
source_type FROM "IMMORTAL"."KATEGORIA" IK ;

INSERT INTO staging.mapa_Kategorie (id_staging,id_base,source_type)
SELECT staging.mapa_Kategorie_seq.nextval,PK.id_Kategorii as id_base ,1 as
source_type FROM "PARAGON"."KATEGORIA" PK;
CREATE TABLE Kategorie
(
    id number(10) PRIMARY KEY,
    nazwa varchar2(64) NOT NULL
);

INSERT INTO Kategorie
(id, nazwa)
SELECT MU.id_staging, PU.Nazwa
FROM "PARAGON"."UZYTKOWNIK" PU LEFT JOIN STAGING.mapa_Kategorie MU ON
PU.ID_KATEGORII = MU.id_base and MU.source_type=1;

INSERT INTO Kategorie
(id, nazwa)
```

```

SELECT MU.id_staging, IU.nazwa
FROM "IMMORTAL"."UZYTKOWNIK" IU LEFT JOIN STAGING.mapa_Kategorie MU ON
IU.ID = MU.id_base and MU.source_type=0;

```

6. Na tym etapie wprowadziliśmy szereg drobnych zmian. Niektóre z nich to:
  - w jednej bazie poprawiona została generacja użytkownika i dodano nowe pole do tabeli *aukcja*. Zmiany te umożliwiły poprawne działanie wymiaru *kategorie* w hurtowni.
  - W Staging Area dodano także pole *idUzytkownikaWygraneego* umożliwiające przypisanie danej aukcji do jej zwycięzcy,
  - oraz pole *statusOferty* dzięki któremu można określić czy dana aukcja zakończyła się wygraniem przez jakiegokolwiek użytkownika.
  
7. Następnie dodaliśmy widok *Kategorie* oraz uzupełniliśmy braki w pozostałych widokach w Hurtowni. Poniżej znajduje się kod prezentujący widoki *Kategorie* oraz *Oferty*:

```

SELECT SK.ID , SK.NAZWA
FROM STAGING.KATEGORIE SK

...

SELECT
STAGING.OFERTY.ID,
CASE WHEN STAGING.WYROZNENIEOFERTY.IDOFERTY is null then 0
Else 1 End
As CZY_OFERTA_WYROZNIONA,
OBlicz_Pore_Dnia_fun(DATAZAMKNIECIA) CZAS_ID
,DATAUTWORZENIA, DATAZAMKNIECIA, LICZBAWSWIETLEN, TYPOFERTY, CENA,
CENAMINIMALNA, STATUSOFERTY, IDKATEGORII, IDLICYTACJI, IDUZYTKOWNIKA,
IDUZYTKOWNIKAwyGRANEGO
FROM STAGING.OFERTY
LEFT JOIN STAGING.WYROZNENIEOFERTY ON STAGING.WYROZNENIEOFERTY.IDOFERTY=
STAGING.OFERTY.ID

```

8. Kolejnym krokiem było utworzenie widoku dla tabeli faktów

```

SELECT H_CZ.CZAS as Kiedy, H_UZ.ID as Kto,H_KA.ID as Kategoria,
CASE WHEN count(H_OF.ID) > 0 then count(H_OF.ID)
Else 0 End
As Ile_Ofert,

sum(cena) as Za_ile
FROM "HURTOWNIA".OFERTY H_OF
RIGHT JOIN HURTOWNIA.CZAS H_CZ on H_OF.CZAS_ID= H_CZ.CZAS

```

```

RIGHT JOIN HURTOWNIA.UZYTKOWNICY H_UZ on H_OF.IDUZYTKOWNIKA = H_UZ.ID
RIGHT JOIN HURTOWNIA.KATEGORIE H_KA on H_OF.IDKATEGORII = H_KA.ID
Group by H_CZ.CZAS,H_UZ.ID, H_KA.ID
order by H_UZ.ID desc

```

9. Po tych zmianach uaktualniliśmy kod tworzący tabelę *Czas w Hurtowni*, dodając obliczanie dnia tygodnia.

```

CREATE TABLE Czas
(
    czas timestamp PRIMARY KEY,
    dzien number,
    dzien_tyg number,
    miesiac number,
    kwartal number,
    rok number
);

Declare
Obecny_czas Timestamp;
Obecny_godzina Number;
Obecny_pora_dnia Number;
Obecny_dzien Number;
Obecny_dzien_tyg Number;
Obecny_miesiac Number;
Obecny_kwartał Number;
Obecny_rok Number;
Koncowy_czas Timestamp :=TO_TIMESTAMP ('2010/12/31 00:00:00.000000',
'YYYY/MM/DD HH24:MI:SS.FF');
Czas_poczatkowy Timestamp := TO_TIMESTAMP ('2000/01/01 00:00:00.000000',
'YYYY/MM/DD HH24:MI:SS.FF');
Ilosc_Wpisow Integer := extract(DAY FROM (Koncowy_czas -Czas_poczatkowy));

BEGIN
FOR i IN 0 .. Ilosc_Wpisow LOOP
    Obecny_czas:=Czas_poczatkowy+numtodsinterval(i*6,'HOUR');
    Obecny_godzina:=extract(HOUR FROM Obecny_czas);
    Obecny_dzien:=extract(DAY FROM Obecny_czas);
    Obecny_miesiac:=extract(MONTH FROM Obecny_czas);
    Obecny_rok:=extract(YEAR FROM Obecny_czas);
    OBLICZ_DZIEN_TYG(Obecny_czas,Obecny_dzien_tyg);
    KWARTAL(Obecny_miesiac,Obecny_kwartał);
    PORA_DNIA(Obecny_pora_dnia,Obecny_godzina);
    INSERT INTO Czas
    (czas,dzien,dzien_tyg,miesiac,kwartał,rok)
    VALUES
    (Obecny_czas,Obecny_dzien,Obecny_dzien_tyg,Obecny_miesiac,Obecny_kwartał,Ob
    ecny_rok);
    END LOOP;
END;

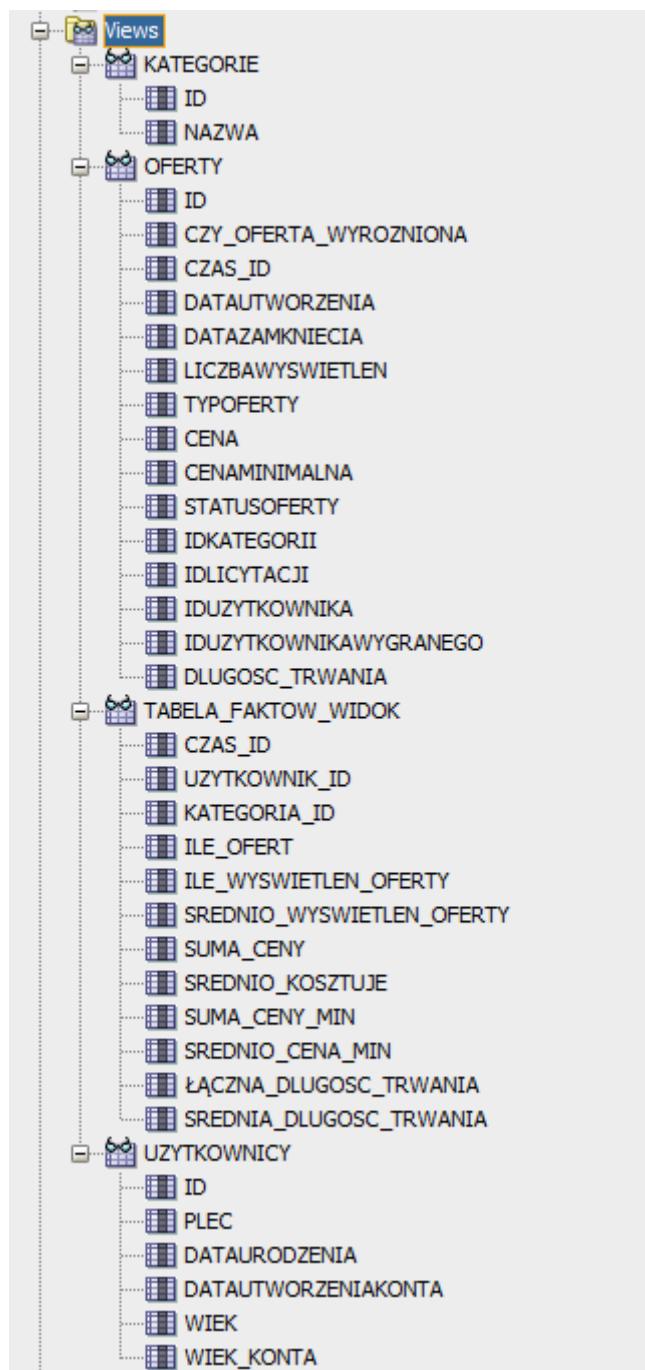
```

**10.** Tabelę faktów w Hurtowni spartycjonowaliśmy dla każdego roku od 2001 do 2011.

```
--Tabla faktow
CREATE TABLE TABELA_FAKTOW
(
CZAS_ID TIMESTAMP(6),
UZYTKOWNIK_ID NUMBER(10),
KATEGORIA_ID NUMBER(10),
--miary
ILE_OFERT NUMBER,
ILE_WYSWIETLEN_OFERTY NUMBER,
SREDNIO_WYSWIETLEN_OFERTY NUMBER,
SUMA_CENY NUMBER,
SREDNIO_KOSZTUJE NUMBER,
ŁACZNA_DLUGOSC_TRWANIA NUMBER,
SREDNIA_DLUGOSC_TRWANIA NUMBER
)
PARTITION BY RANGE (CZAS_ID)
(
PARTITION sales_2000 VALUES LESS THAN (TO_TIMESTAMP ('2001/01/01
00:00:00.000000', 'YYYY/MM/DD HH24:MI:SS.FF')) TABLESPACE tbs_2000,
PARTITION sales_2001 VALUES LESS THAN (TO_TIMESTAMP ('2002/01/01
00:00:00.000000', 'YYYY/MM/DD HH24:MI:SS.FF')) TABLESPACE tbs_2001,
PARTITION sales_2002 VALUES LESS THAN (TO_TIMESTAMP ('2003/01/01
00:00:00.000000', 'YYYY/MM/DD HH24:MI:SS.FF')) TABLESPACE tbs_2002,
PARTITION sales_2003 VALUES LESS THAN (TO_TIMESTAMP ('2004/01/01
00:00:00.000000', 'YYYY/MM/DD HH24:MI:SS.FF')) TABLESPACE tbs_2003,
PARTITION sales_2004 VALUES LESS THAN (TO_TIMESTAMP ('2005/01/01
00:00:00.000000', 'YYYY/MM/DD HH24:MI:SS.FF')) TABLESPACE tbs_2004,
PARTITION sales_2005 VALUES LESS THAN (TO_TIMESTAMP ('2006/01/01
00:00:00.000000', 'YYYY/MM/DD HH24:MI:SS.FF')) TABLESPACE tbs_2005,
PARTITION sales_2006 VALUES LESS THAN (TO_TIMESTAMP ('2007/01/01
00:00:00.000000', 'YYYY/MM/DD HH24:MI:SS.FF')) TABLESPACE tbs_2006,
PARTITION sales_2007 VALUES LESS THAN (TO_TIMESTAMP ('2008/01/01
00:00:00.000000', 'YYYY/MM/DD HH24:MI:SS.FF')) TABLESPACE tbs_2007,
PARTITION sales_2008 VALUES LESS THAN (TO_TIMESTAMP ('2009/01/01
00:00:00.000000', 'YYYY/MM/DD HH24:MI:SS.FF')) TABLESPACE tbs_2008,
PARTITION sales_2009 VALUES LESS THAN (TO_TIMESTAMP ('2010/01/01
00:00:00.000000', 'YYYY/MM/DD HH24:MI:SS.FF')) TABLESPACE tbs_2009,
PARTITION sales_2010 VALUES LESS THAN (TO_TIMESTAMP ('2011/01/01
00:00:00.000000', 'YYYY/MM/DD HH24:MI:SS.FF')) TABLESPACE tbs_2010
);
```

Po tych zmianach byliśmy gotowi by zacząć tworzyć ostatecznie zapytania.

### 3. Widoki



Aby ułatwić tworzenie zapytań w *Hurtowni* przygotowaliśmy odpowiednie widoki dla każdego wymiaru oraz dla tabeli faktów. Poniżej znajdują się wyjaśnienia niektórych kolumn:

- **OFERTY**
  - CZY\_OFERTA\_WYROZNIONA – wartość 1 oznacza ofertę wyróżnioną, wartość 0 – nie.
  - CZAS\_ID – czas obliczony przez procedurę wykorzystywany do wyznaczenia pory dnia. Do obliczeń wykorzystujemy datę zakończenia oferty.
  - IDKATEGORII – zawiera ID kategorii, do której należy dana oferta

- IDUZYTKOWNIKA – zawiera ID osoby, która utworzyła aukcję
  - IDUZYTKOWNIKAWYGRANEGO – zawiera ID osoby, która wygrała aukcję
  - DLUGOSCTRWANIA – jest to ilość dni od momentu utworzenia aukcji do jej zamknięcia
- **TABELA\_FAKTOW\_WIDOK**
  - **Klucz główny**
    - CZAS\_ID – klucz obcy do tablicy CZAS.
    - UZYTKOWNIK\_ID – klucz obcy do widoku Uzytkownicy
    - KATEGORIA\_ID – klucz obcy do widoku Kategorie
  - **Miary**
    - ILE\_OFERT – Ile ofert było wystawionych w danym okresie czasu, przez danego użytkownika w danej kategorii.
    - ILE\_WYSWIETLEN\_OFERTY – Ile było wyświetleń ofert wystawionych w danym okresie czasu, przez danego użytkownika w danej kategorii.
    - SREDNIO\_WYSWIETLEN\_OFERTY – średnia ilości wyświetleń ofert wystawionych w danym okresie czasu, przez danego użytkownika w danej kategorii.
    - SUMA\_CENY – łączna cena ofert wystawionych w danym okresie czasu, przez danego użytkownika w danej kategorii.
    - SREDNIO\_KOSZTUJE – średnia cena ofert wystawionych w danym okresie czasu, przez danego użytkownika w danej kategorii.
    - SUMA\_CENY\_MIN – łączna cena minimalna ofert wystawionych w danym okresie czasu, przez danego użytkownika w danej kategorii.
    - SCREDNIA\_CENY\_MIN – średnia cena minimalna ofert wystawionych w danym okresie czasu, przez danego użytkownika w danej kategorii.
    - ŁĄCZNA\_DLUGOSC\_TRWANIA – łączna długość trwania ofert wystawionych w danym okresie czasu, przez danego użytkownika w danej kategorii.
    - SREDNIA\_DLUGOSC\_TRWANIA – średnia długość trwania ofert wystawionych w danym okresie czasu, przez danego użytkownika w danej kategorii.

TABELA\_FAKTOW\_WIDOK – była wykorzystywana podczas projektowania tablicy TABELA\_FAKTOW i jest jej dokładnym odpowiednikiem

- **UZYTKOWNICY**
  - WIEK\_KONTA – liczba lat liczona od momentu utworzenia konta do daty utworzenia wpisu w bazie

## 4. Zapytania

### 1. Informacje o kupowaniu z podziałem na płci

```
select sum(TF.ILE_OFERT) as Lacznie_Ofert,
sum(CASE WHEN H_UZ.PLEC = '0' THEN TF.ILE_OFERT ELSE NULL END) AS
Wystawili_Kobiety,
sum(CASE WHEN H_UZ.PLEC = '1' THEN TF.ILE_OFERT ELSE NULL END) AS
Wystawili_Męzczyźni,
count((SELECT H_UZ_WY.PLEC from hurtownia.uzytkownicy H_UZ_WY where
H_UZ_WY.ID=H_OF.IDUZYTKOWNIKAWYGRANEGO and H_UZ_WY.PLEC='0' )) AS
Kupili_Kobiety,
count((SELECT H_UZ_WY.PLEC from hurtownia.uzytkownicy H_UZ_WY where
H_UZ_WY.ID=H_OF.IDUZYTKOWNIKAWYGRANEGO and H_UZ_WY.PLEC='1' )) AS
Kupili_Męczyszni,
count(CASE WHEN H_OF.IDUZYTKOWNIKAWYGRANEGO IS NULL THEN 1 ELSE NULL END)
AS Nie_Kupione
FROM TABELA_FAKTOW TF
LEFT JOIN hurtownia.czas H_CZ on H_CZ.CZAS = TF.CZAS_ID
LEFT JOIN hurtownia.uzytkownicy H_UZ on H_UZ.ID = TF.UZYTKOWNIK_ID
LEFT JOIN hurtownia.kategorie H_K on H_K.ID = TF.KATEGORIA_ID
LEFT JOIN hurtownia.oferty H_OF on H_OF.IDUZYTKOWNIKA = TF.UZYTKOWNIK_ID;
```

### 2. Ile ofert zostało wystawionych w poszczególnych latach, z podziałem na płci

```
select
--2000
sum(CASE WHEN H_CZ.ROK=2000 THEN TF.ILE_OFERT ELSE NULL END) as
Lacznie_Ofert_2000,
sum(CASE WHEN H_UZ.PLEC = '0' AND H_CZ.ROK=2000 THEN TF.ILE_OFERT ELSE NULL
END) AS Wystawili_Kobiety_2000,
sum(CASE WHEN H_UZ.PLEC = '1' AND H_CZ.ROK=2000 THEN TF.ILE_OFERT ELSE NULL
END) AS Wystawili_Męzczyźni_2000,
--2001
sum(CASE WHEN H_CZ.ROK=2001 THEN TF.ILE_OFERT ELSE NULL END) as
Lacznie_Ofert_2001,
sum(CASE WHEN H_UZ.PLEC = '0' AND H_CZ.ROK=2001 THEN TF.ILE_OFERT ELSE NULL
END) AS Wystawili_Kobiety_2001,
sum(CASE WHEN H_UZ.PLEC = '1' AND H_CZ.ROK=2001 THEN TF.ILE_OFERT ELSE NULL
END) AS Wystawili_Męzczyźni_2001,
--2002
sum(CASE WHEN H_CZ.ROK=2002 THEN TF.ILE_OFERT ELSE NULL END) as
Lacznie_Ofert_2002,
sum(CASE WHEN H_UZ.PLEC = '0' AND H_CZ.ROK=2002 THEN TF.ILE_OFERT ELSE NULL
END) AS Wystawili_Kobiety_2002,
sum(CASE WHEN H_UZ.PLEC = '1' AND H_CZ.ROK=2002 THEN TF.ILE_OFERT ELSE NULL
END) AS Wystawili_Męzczyźni_2002,
--2003
sum(CASE WHEN H_CZ.ROK=2003 THEN TF.ILE_OFERT ELSE NULL END) as
Lacznie_Ofert_2003,
sum(CASE WHEN H_UZ.PLEC = '0' AND H_CZ.ROK=2003 THEN TF.ILE_OFERT ELSE NULL
END) AS Wystawili_Kobiety_2003,
sum(CASE WHEN H_UZ.PLEC = '1' AND H_CZ.ROK=2003 THEN TF.ILE_OFERT ELSE NULL
END) AS Wystawili_Męzczyźni_2003,
```

```

--2004
sum(CASE WHEN H_CZ.ROK=2004 THEN TF.ILE_OFERT ELSE NULL END) as
Lacznie_Ofert_2004,
sum(CASE WHEN H_UZ.PLEC = '0' AND H_CZ.ROK=2004 THEN TF.ILE_OFERT ELSE NULL
END) AS Wystawili_Kobiety_2004,
sum(CASE WHEN H_UZ.PLEC = '1' AND H_CZ.ROK=2004 THEN TF.ILE_OFERT ELSE NULL
END) AS Wystawili_Męzczyzni_2004,
--2005
sum(CASE WHEN H_CZ.ROK=2005 THEN TF.ILE_OFERT ELSE NULL END) as
Lacznie_Ofert_2005,
sum(CASE WHEN H_UZ.PLEC = '0' AND H_CZ.ROK=2005 THEN TF.ILE_OFERT ELSE NULL
END) AS Wystawili_Kobiety_2005,
sum(CASE WHEN H_UZ.PLEC = '1' AND H_CZ.ROK=2005 THEN TF.ILE_OFERT ELSE NULL
END) AS Wystawili_Męzczyzni_2005,
--2006
sum(CASE WHEN H_CZ.ROK=2006 THEN TF.ILE_OFERT ELSE NULL END) as
Lacznie_Ofert_2006,
sum(CASE WHEN H_UZ.PLEC = '0' AND H_CZ.ROK=2006 THEN TF.ILE_OFERT ELSE NULL
END) AS Wystawili_Kobiety_2006,
sum(CASE WHEN H_UZ.PLEC = '1' AND H_CZ.ROK=2006 THEN TF.ILE_OFERT ELSE NULL
END) AS Wystawili_Męzczyzni_2006,
--2007
sum(CASE WHEN H_CZ.ROK=2007 THEN TF.ILE_OFERT ELSE NULL END) as
Lacznie_Ofert_2007,
sum(CASE WHEN H_UZ.PLEC = '0' AND H_CZ.ROK=2007 THEN TF.ILE_OFERT ELSE NULL
END) AS Wystawili_Kobiety_2007,
sum(CASE WHEN H_UZ.PLEC = '1' AND H_CZ.ROK=2007 THEN TF.ILE_OFERT ELSE NULL
END) AS Wystawili_Męzczyzni_2007,
--2008
sum(CASE WHEN H_CZ.ROK=2008 THEN TF.ILE_OFERT ELSE NULL END) as
Lacznie_Ofert_2008,
sum(CASE WHEN H_UZ.PLEC = '0' AND H_CZ.ROK=2008 THEN TF.ILE_OFERT ELSE NULL
END) AS Wystawili_Kobiety_2008,
sum(CASE WHEN H_UZ.PLEC = '1' AND H_CZ.ROK=2008 THEN TF.ILE_OFERT ELSE NULL
END) AS Wystawili_Męzczyzni_2008,
--2009
sum(CASE WHEN H_CZ.ROK=2009 THEN TF.ILE_OFERT ELSE NULL END) as
Lacznie_Ofert_2009,
sum(CASE WHEN H_UZ.PLEC = '0' AND H_CZ.ROK=2009 THEN TF.ILE_OFERT ELSE NULL
END) AS Wystawili_Kobiety_2009,
sum(CASE WHEN H_UZ.PLEC = '1' AND H_CZ.ROK=2009 THEN TF.ILE_OFERT ELSE NULL
END) AS Wystawili_Męzczyzni_2009,
--2010
sum(CASE WHEN H_CZ.ROK=2010 THEN TF.ILE_OFERT ELSE NULL END) as
Lacznie_Ofert_2010,
sum(CASE WHEN H_UZ.PLEC = '0' AND H_CZ.ROK=2010 THEN TF.ILE_OFERT ELSE NULL
END) AS Wystawili_Kobiety_2010,
sum(CASE WHEN H_UZ.PLEC = '1' AND H_CZ.ROK=2010 THEN TF.ILE_OFERT ELSE NULL
END) AS Wystawili_Męzczyzni_2010

FROM TABELA_FAKTOW TF
LEFT JOIN hurtownia.czas H_CZ on H_CZ.CZAS = TF.CZAS_ID
LEFT JOIN hurtownia.uzytkownicy H_UZ on H_UZ.ID = TF.UZYTKOWNIK_ID
LEFT JOIN hurtownia.kategorie H_K on H_K.ID = TF.KATEGORIA_ID
LEFT JOIN hurtownia.oferty H_OF on H_OF.IDUZYTKOWNIKA = TF.UZYTKOWNIK_ID

```

3. Ile średnio wydają w grudniu mężczyźni a ile kobiety, z podzielam na grupy wiekowe  
0-10, 10-20, 20-30, 30-50, >50

```

select
AVG(CASE WHEN H_UZ.PLEC = '0' AND H_UZ.WIEK<10 THEN TF.SUMA_CENY ELSE NULL
END) AS Sr_Wy_Kobiety_0_9,
AVG(CASE WHEN H_UZ.PLEC = '1' AND H_UZ.WIEK<10 THEN TF.SUMA_CENY ELSE NULL
END) AS Sr_Wy_Mężczyzn_0_9,
AVG(CASE WHEN H_UZ.PLEC = '0' AND H_UZ.WIEK<20 and H_UZ.WIEK>10 THEN
TF.SUMA_CENY ELSE NULL END) AS Sr_Wy_Kobiety_10_19,
AVG(CASE WHEN H_UZ.PLEC = '1' AND H_UZ.WIEK<20 and H_UZ.WIEK>10 THEN
TF.SUMA_CENY ELSE NULL END) AS Sr_Wy_Mężczyzn_10_19,
AVG(CASE WHEN H_UZ.PLEC = '0' AND H_UZ.WIEK<30 and H_UZ.WIEK>20 THEN
TF.SUMA_CENY ELSE NULL END) AS Sr_Wy_Kobiety_20_29,
AVG(CASE WHEN H_UZ.PLEC = '1' AND H_UZ.WIEK<30 and H_UZ.WIEK>20 THEN
TF.SUMA_CENY ELSE NULL END) AS Sr_Wy_Mężczyzn_20_29,
AVG(CASE WHEN H_UZ.PLEC = '0' AND H_UZ.WIEK<40 and H_UZ.WIEK>30 THEN
TF.SUMA_CENY ELSE NULL END) AS Sr_Wy_Kobiety_30_39,
AVG(CASE WHEN H_UZ.PLEC = '1' AND H_UZ.WIEK<40 and H_UZ.WIEK>30 THEN
TF.SUMA_CENY ELSE NULL END) AS Sr_Wy_Mężczyzn_30_39,
AVG(CASE WHEN H_UZ.PLEC = '0' AND H_UZ.WIEK<50 and H_UZ.WIEK>40 THEN
TF.SUMA_CENY ELSE NULL END) AS Sr_Wy_Kobiety_40_49,
AVG(CASE WHEN H_UZ.PLEC = '1' AND H_UZ.WIEK<50 and H_UZ.WIEK>40 THEN
TF.SUMA_CENY ELSE NULL END) AS Sr_Wy_Mężczyzn_40_49,
AVG(CASE WHEN H_UZ.PLEC = '0' AND H_UZ.WIEK>50 THEN TF.SUMA_CENY ELSE NULL
END) AS Sr_Wy_Kobiety_50,
AVG(CASE WHEN H_UZ.PLEC = '1' AND H_UZ.WIEK>50 THEN TF.SUMA_CENY ELSE
NULL END) AS Sr_Wy_Mężczyzn_50
FROM TABELA_FAKTOW TF
LEFT JOIN hurtownia.czas H_CZ on H_CZ.CZAS = TF.CZAS_ID
LEFT JOIN hurtownia.uzytkownicy H_UZ on H_UZ.ID = TF.UZYTKOWNIK_ID
LEFT JOIN hurtownia.kategorie H_K on H_K.ID = TF.KATEGORIA_ID
WHERE H_CZ.Miesiac=12;

```

4. W jakich porach dnia, ile jest wygrywanych aukcji, z podziałem na płci

```

select
--od 0 do 6
SUM(CASE WHEN H_UZ.PLEC = '0' AND OBLICZ_PORE_DNIA_NUM_FUN(H_CZ.CZAS)=0
THEN TF.ILE_OFERT ELSE NULL END) AS Wy_Kobiety_0_6,
SUM(CASE WHEN H_UZ.PLEC = '1' AND OBLICZ_PORE_DNIA_NUM_FUN(H_CZ.CZAS)=0
THEN TF.ILE_OFERT ELSE NULL END) AS Wy_Mężczyzn_0_6,
--od 6 do 12
SUM(CASE WHEN H_UZ.PLEC = '0' AND OBLICZ_PORE_DNIA_NUM_FUN(H_CZ.CZAS)=1
THEN TF.ILE_OFERT ELSE NULL END) AS Wy_Kobiety_6_12,
SUM(CASE WHEN H_UZ.PLEC = '1' AND OBLICZ_PORE_DNIA_NUM_FUN(H_CZ.CZAS)=1
THEN TF.ILE_OFERT ELSE NULL END) AS Wy_Mężczyzn_6_12,
--od 12 do 18
SUM(CASE WHEN H_UZ.PLEC = '0' AND OBLICZ_PORE_DNIA_NUM_FUN(H_CZ.CZAS)=2
THEN TF.ILE_OFERT ELSE NULL END) AS Wy_Kobiety_12_18,
SUM(CASE WHEN H_UZ.PLEC = '1' AND OBLICZ_PORE_DNIA_NUM_FUN(H_CZ.CZAS)=2
THEN TF.ILE_OFERT ELSE NULL END) AS Wy_Mężczyzn_12_18,
--od 18 do 24
SUM(CASE WHEN H_UZ.PLEC = '0' AND OBLICZ_PORE_DNIA_NUM_FUN(H_CZ.CZAS)=3
THEN TF.ILE_OFERT ELSE NULL END) AS Wy_Kobiety_18_24,
SUM(CASE WHEN H_UZ.PLEC = '1' AND OBLICZ_PORE_DNIA_NUM_FUN(H_CZ.CZAS)=3
THEN TF.ILE_OFERT ELSE NULL END) AS Wy_Mężczyzn_18_24

```

```

FROM TABELA_FAKTOW TF
LEFT JOIN hurtownia.czas H_CZ on H_CZ.CZAS = TF.CZAS_ID
LEFT JOIN hurtownia.uzytkownicy H_UZ on H_UZ.ID = TF.UZYTKOWNIK_ID
LEFT JOIN hurtownia.kategorie H_K on H_K.ID = TF.KATEGORIA_ID

```

5. Jaki jest średni czas trwania oferty, z podziałem na płci

```

select sum(TF.ILE_OFERT) as Lacznie_Ofert,
avg(CASE WHEN H_UZ.PLEC = '0' THEN TF.SREDNIA_DLUGOSC_TRWANIA ELSE NULL
END) AS Srednio_Trwaja_Kobiety,
avg(CASE WHEN H_UZ.PLEC = '1' THEN TF.SREDNIA_DLUGOSC_TRWANIA ELSE NULL
END) AS Srednio_Trwaja_Męzczyzni,

FROM TABELA_FAKTOW TF
LEFT JOIN hurtownia.czas H_CZ on H_CZ.CZAS = TF.CZAS_ID
LEFT JOIN hurtownia.uzytkownicy H_UZ on H_UZ.ID = TF.UZYTKOWNIK_ID
LEFT JOIN hurtownia.kategorie H_K on H_K.ID = TF.KATEGORIA_ID
LEFT JOIN hurtownia.oferty H_OF on H_OF.IDUZYTKOWNIKA = TF.UZYTKOWNIK_ID

```

6. Jakie średnie ceny minimalne miały oferty wystawiane w latach 2005-2008 dla kategorii o id = 2

```

select
--2005
avg(CASE WHEN H_CZ.ROK=2005 THEN TF.SREDNIA_CENY_MIN ELSE NULL END) as
Srednio_Rok_2005,
avg(CASE WHEN H_UZ.PLEC = '0' AND H_CZ.ROK=2005 THEN TF.SREDNIA_CENY_MIN
ELSE NULL END) AS Srednio_Kobiety_2005,
avg(CASE WHEN H_UZ.PLEC = '1' AND H_CZ.ROK=2005 THEN TF.SREDNIA_CENY_MIN
ELSE NULL END) AS Srednio_Męzczyzni_2005,
--2006
avg(CASE WHEN H_CZ.ROK=2006 THEN TF.SREDNIA_CENY_MIN ELSE NULL END) as
Srednio_Rok_2006,
avg(CASE WHEN H_UZ.PLEC = '0' AND H_CZ.ROK=2006 THEN TF.SREDNIA_CENY_MIN
ELSE NULL END) AS Srednio_Kobiety_2006,
avg(CASE WHEN H_UZ.PLEC = '1' AND H_CZ.ROK=2006 THEN TF.SREDNIA_CENY_MIN
ELSE NULL END) AS Srednio_Męzczyzni_2006,
--2007
avg(CASE WHEN H_CZ.ROK=2007 THEN TF.SREDNIA_CENY_MIN ELSE NULL END) as
Srednio_Rok_2007,
avg(CASE WHEN H_UZ.PLEC = '0' AND H_CZ.ROK=2007 THEN TF.SREDNIA_CENY_MIN
ELSE NULL END) AS Srednio_Kobiety_2007,
avg(CASE WHEN H_UZ.PLEC = '1' AND H_CZ.ROK=2007 THEN TF.SREDNIA_CENY_MIN
ELSE NULL END) AS Srednio_Męzczyzni_2007,
--2008
avg(CASE WHEN H_CZ.ROK=2008 THEN TF.SREDNIA_CENY_MIN ELSE NULL END) as
Srednio_Rok_2008,

```

```

avg(CASE WHEN H_UZ.PLEC = '0' AND H_CZ.ROK=2008 THEN TF.SREDNIA_CENY_MIN
ELSE NULL END) AS Srednio_Kobiety_2008,
avg(CASE WHEN H_UZ.PLEC = '1' AND H_CZ.ROK=2008 THEN TF.SREDNIA_CENY_MIN
ELSE NULL END) AS Srednio_Męzczyzni_2008

FROM TABELA_FAKTOW TF
LEFT JOIN hurtownia.czas H_CZ on H_CZ.CZAS = TF.CZAS_ID
LEFT JOIN hurtownia.uzytkownicy H_UZ on H_UZ.ID = TF.UZYTKOWNIK_ID
LEFT JOIN hurtownia.kategorie H_K on H_K.ID = TF.KATEGORIA_ID
LEFT JOIN hurtownia.oferty H_OF on H_OF.IDUZYTKOWNIKA = TF.UZYTKOWNIK_ID
WHERE H_K.ID=2 ;

```

7. Ile średnio wyświetleń mają oferty z ceną minimalną mniejszą od 500zł w porównaniu do reszty ofert

```

SELECT
avg(CASE WHEN H_OF.CENA>=500 THEN TF.SREDNIO_WYSWIETLEN_OFERTY ELSE NULL
END) as WIEKSZE_OD_500,
avg(CASE WHEN H_OF.CENA < 500 THEN TF.SREDNIO_WYSWIETLEN_OFERTY ELSE NULL
END) AS MNIEJSZE_OD_500

FROM TABELA_FAKTOW TF
LEFT JOIN hurtownia.czas H_CZ on H_CZ.CZAS = TF.CZAS_ID
LEFT JOIN hurtownia.uzytkownicy H_UZ on H_UZ.ID = TF.UZYTKOWNIK_ID
LEFT JOIN hurtownia.kategorie H_K on H_K.ID = TF.KATEGORIA_ID
LEFT JOIN hurtownia.oferty H_OF on H_OF.IDUZYTKOWNIKA = TF.UZYTKOWNIK_ID;

```

8. Ile użytkowników z kontami starszymi od 5 lat nie wystawiło żadnej oferty

```

SELECT
count(CASE WHEN TF.UZYTKOWNIK_ID IS NULL THEN 1 ELSE NULL END) as
Łacznie_Nie_Wystawilo,
count(CASE WHEN TF.UZYTKOWNIK_ID IS NULL AND H_UZ.PLEC = '0' THEN 1 ELSE
NULL END) AS Nie_Wystawilo_Kobiety,
count(CASE WHEN TF.UZYTKOWNIK_ID IS NULL AND H_UZ.PLEC = '1' THEN 1 ELSE
NULL END) AS Nie_Wystawilo_Męzczyzni,
count(CASE WHEN TF.UZYTKOWNIK_ID IS NOT NULL THEN 1 ELSE NULL END) as
Łacznie_Wystawilo,
count(CASE WHEN TF.UZYTKOWNIK_ID IS NOT NULL AND H_UZ.PLEC = '0' THEN 1
ELSE NULL END) AS Wystawilo_Kobiety,
count(CASE WHEN TF.UZYTKOWNIK_ID IS NOT NULL AND H_UZ.PLEC = '1' THEN 1
ELSE NULL END) AS Wystawilo_Męzczyzni
FROM TABELA_FAKTOW TF
RIGHT JOIN hurtownia.uzytkownicy H_UZ on H_UZ.ID = TF.UZYTKOWNIK_ID
WHERE H_UZ.WIEK_KONTA >5 ;

-- 
select sum(TF.ILE_WYSWIETLEN_OFERTY) as Laczniewyswietlen,
avg(TF.ILE_WYSWIETLEN_OFERTY) as Sredniewyswietlen_Na_Oferete
from TABELA_FAKTOW TF
LEFT JOIN hurtownia.czas H_CZ on H_CZ.CZAS = TF.CZAS_ID
LEFT JOIN hurtownia.uzytkownicy H_UZ on H_UZ.ID = TF.UZYTKOWNIK_ID
LEFT JOIN hurtownia.kategorie H_K on H_K.ID = TF.KATEGORIA_ID
WHERE H_UZ.WIEK < 25 and H_CZ.Miesiac=1 and H_K.ID=1;

```