Sveučilište J. J. Strossmayera u Osijeku
Fakultet primijenjene matematike i informatike

Seminarski rad

GLAZBENI STREAMING SERVIS

Moderni sustavi baza podataka

Leopold Brodar

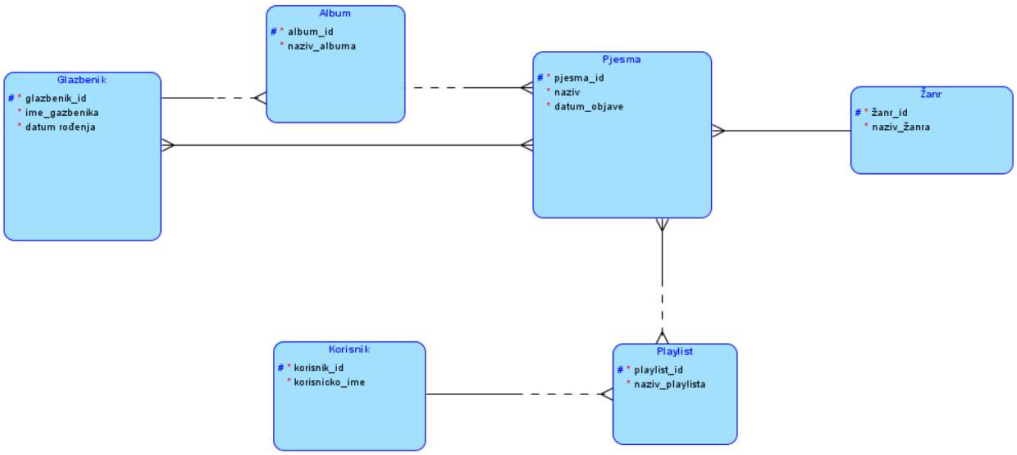
Osijek, 2023.

SADRŽAJ

UVOD

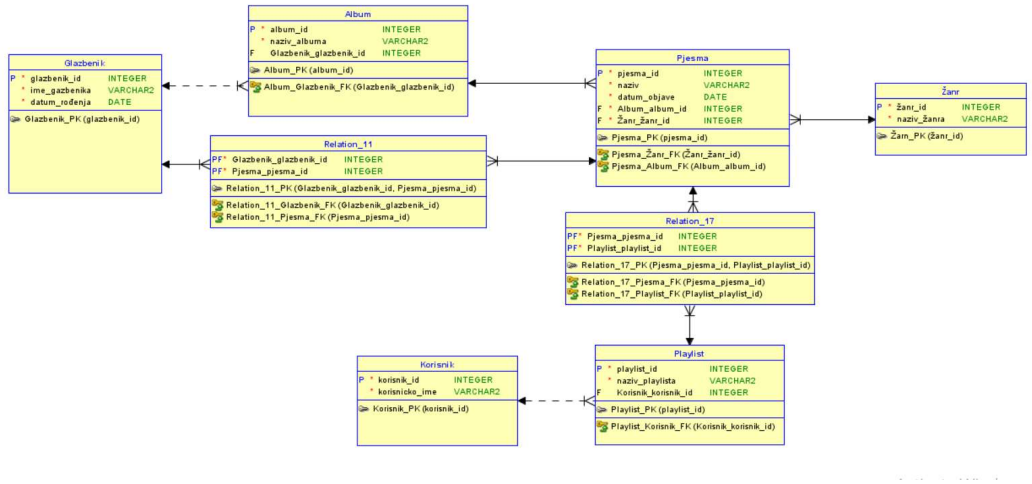
Glazbeni streaming servisi su digitalne platforme koje omogućuju korisnicima pristup ogromnom katalogu glazbenih sadržaja putem interneta. Ovi servisi revolucionirali su način na koji slušamo glazbu, nudeći neograničen pristup milijunima pjesama bez potrebe za preuzimanjem. Ovaj seminar fokusira se na bazu podataka takvih servisa.

MODEL ENTITETA I VEZA



Krenimo od najvažnijeg entieta što je pjesma. Svaka pjesma ima žanr, više pjesama mogu biti istog žanra. Pjesma mora imati jednog ili više autora(glazbenika). Svaki glazbenik mora imati jednu ili više pjesama. Glazbenik može ali ne mora objaviti album. Album sadrži više pjesama, ali svaka pjesma ne treba pripadati albumu. Pjesma može biti u više playlista. Playlist sadržava više pjesama. Korisnik može napraviti više playlista.

RELACIJSKI MODEL



Iz prethodnog MEV modela dobivamo dani relacijski model koristeći pravila za pretvorbu MEV modela u relacijski model. Nadodane tablice su nastale pretvorbom više prema više veza.

TABLICE

```
CREATE TABLE PJESMA(
    PJESMA_ID INTEGER PRIMARY KEY,
    NAZIV_PJESME VARCHAR2(100) NOT NULL,
    DATUM_OBJAVE DATE NOT NULL,
    ZANR_ID INTEGER NOT NULL REFERENCES ZANR(ZANR_ID),
    ALBUM_ID INTEGER REFERENCES ALBUM(ALBUM_ID)
);
```

Prikazati ću kreaciu najvažnije tablice pjesma. Primarni ključ tablice je PJESMA_ID. Tablica još ima atribute NAZIV_PJESME i DATUM_OBJAVE kojima naziv objašnjava svrhu. ZANR_ID povezuje pjesmu s njenim zanrom, a ALBUM_ID sa albumom ako se u njemu nalazi. Kreirane su i sve ostale tablice.

UNOSI

Nakon kreiranja tablica bazu možemo početi popunjavati podacima. Prikazati ću nekoliko primjera unosa podataka:

```
INSERT INTO PJESMA VALUES (
    1,
    'Hello',
    TO_DATE('20/11/2015', 'dd/mm/yyyy'),
    1,
    1
);

INSERT INTO PJESMA VALUES (
    2,
    'Send My Love (To Your New Lover)',
    TO_DATE('20/11/2015', 'dd/mm/yyyy'),
    1,
    1
);
```

```
INSERT INTO GLAZBENIK(
    GLAZBENIK_ID,
    IME_GLAZBENIKA
) VALUES (
    1,
    'Adele'
);

INSERT INTO GLAZBENIK(
    GLAZBENIK_ID,
    IME_GLAZBENIKA
) VALUES (
    2,
    'Tame Impala'
);
```

```
INSERT INTO ZANR(
    ZANR_ID,
    NAZIV_ZANRA
) VALUES(
    1,
    'Pop'
);
```

```
INSERT INTO ALBUM(
    ALBUM_ID,
    NAZIV_ALBUMA,
    GLAZBENIK_ID
) VALUES (
    1,
    '25',
    1
);
```

```
INSERT INTO GLAZBENIK_PJESMA VALUES (
    1,
    2
);
```

```
INSERT INTO KORISNIK VALUES(
    1,
    'LunaStar27'
);
```

UPITI

Nakon što smo bazu popunili podacima možemo im pristupiti pomoću upita.

Prikazati ću primjere nekoliko upita.

Ispis imena svih glazbenika:

```
SELECT
    IME_GLAZBENIKA
FROM
    GLAZBENIK;
```

Ispis svih podataka o pjesmama glazbenika Ariana Grande:

```
SELECT
    *
FROM
    PJESMA
    JOIN GLAZBENIK_PJESMA
    ON PJESMA.PJESMA_ID= GLAZBENIK_PJESMA.PJESMA_ID JOIN GLAZBENIK
    ON GLAZBENIK_PJESMA.GLAZBENIK_ID = GLAZBENIK.GLAZBENIK_ID
WHERE
    GLAZBENIK.IME_GLAZBENIKA = 'Ariana Grande';
```

Ispis imena albuma i broja pjesama u njemu:

```
SELECT
    ALBUM.NAZIV_ALBUMA,
    COUNT(PJESMA.PJESMA_ID)
FROM
    PJESMA
    JOIN ALBUM
    ON ALBUM.ALBUM_ID = PJESMA.ALBUM_ID
GROUP BY
    ALBUM.NAZIV_ALBUMA;
```

Ispis svih pjesama koje nisu ni u jednom playlistu:

```
SELECT
    PJESMA.PJESMA_ID,
    PJESMA.NAZIV_PJESME
FROM
    PJESMA
WHERE
    PJESMA.PJESMA_ID NOT IN (
        SELECT
            PJESMA.PJESMA_ID
        FROM
            PJESMA
            JOIN PLAYLIST_PJESMA
            ON PJESMA.PJESMA_ID = PLAYLIST_PJESMA.PJESMA_ID JOIN PLAYLIST
            ON PLAYLIST_PJESMA.PLAYLIST_ID = PLAYLIST.PLAYLIST_ID
    );
```

Ispis svih glazbenika i korisnika:

```
SELECT
    GLAZBENIK.GLAZBENIK_ID,
    GLAZBENIK.IME_GLAZBENIKA,
    'GLAZBENIK'                AS "OPIS"
FROM
    GLAZBENIK UNION
    SELECT
        KORISNIK.KORISNIK_ID,
        KORISNIK.KORISNICKO_IME,
        'KORISNIK'                AS "OPIS"
    FROM
        KORISNIK;
```

ZADANE VRIJEDNOSTI, UVJETI, KOMENTARI I INDEKSI

Zadane vrijednosti su zapravo vrijednosti koje pridodajemo ne nužnim atributima ukoliko ih korisnik ne unese.
Uvjeti nam služe kako bismo mogli onemogućiti unos krivih podataka u tablice od strane korisnika.

```
ALTER TABLE PJESMA MODIFY ALBUM_ID DEFAULT NULL;

ALTER TABLE PJESMA ADD CONSTRAINT DATUM_CHECK CHECK(DATUM_OBJAVE >= TO_DATE('01/01/1900', 'dd/mm/yyyy'));
```

Komentarima dodajemo kratke opise tablica kao što vidimo u danom primjeru.

```
comment on table pjesma is 'Tablica svih pjesama';
comment on table GLAZBENIK is 'Tablica svih glazbenika';
comment on table PLAYLIST is 'Tablica svih playlista';
comment on table zanr is 'Tablica svih zanra';
comment on table korisnik is 'Tablica svih korisnika';
```

Indeksi nam služe za brži pristup često korištenim podacima unutar baze podataka.

```
create INDEX i_pjesma on pjesma(NAZIV_PJESME);
create INDEX i_playlist on playlist(NAZIV_PLAYLISTA);
```

Možemo ih iskoristiti pri pretraživanju pjesama po nazivu.

PROCEDURE

Procedure su skupine naredbi koje se sastavljaju i izvršavaju kao jedna logička jedinica.

```
CREATE OR REPLACE PROCEDURE ADD_TO_PLAYLIST(
    n_PLAYLIST_ID IN PLAYLIST.PLAYLIST_ID%TYPE,
    n_PJESMA_ID IN PJESMA.PJESMA_ID%TYPE
)AS BEGIN
    INSERT INTO PLAYLIST_PJESMA (PLAYLIST_ID, PJESMA_ID) VALUES
    (
        n_PLAYLIST_ID,
        n_PJESMA_ID
    );
    COMMIT;
    EXCEPTION WHEN OTHERS THEN
    ROLLBACK;
    raise_application_error(-20001,'ADD_TO_PLAYLIST ERROR');
END;
/

call ADD_TO_PLAYLIST(1,13);

CREATE OR REPLACE PROCEDURE DELETE_FROM_PLAYLIST(
    n_PLAYLIST_ID IN PLAYLIST.PLAYLIST_ID%TYPE,
    n_PJESMA_ID IN PJESMA.PJESMA_ID%TYPE
)AS BEGIN
    DELETE FROM PLAYLIST_PJESMA
    WHERE PLAYLIST_PJESMA.PLAYLIST_ID = n_PLAYLIST_ID AND PLAYLIST_PJESMA.PJESMA_ID = n_PJESMA_ID;
    COMMIT;
    EXCEPTION WHEN OTHERS THEN
    ROLLBACK;
    raise_application_error(-20001,'DELETE_FROM_PLAYLIST ERROR');
END;
/

cALL DELETE_FROM_PLAYLIST(1,13);
```

Procedura ADD_TO_PLAYLIST omogućuje dodavanje pjesama na neki playlist.

Procedura DELETE_FROM_PLAYLIST omogućuje brisanje pjesama sa playlista.

OKIDAČI

Okidači nam služe kao aktivni elementi koji se aktiviraju prilikom naredbi umetanja, brisanja ili ažuriranja.

```
CREATE OR REPLACE TRIGGER DATUM_CHECK BEFORE INSERT OR UPDATE OF DATUM_OBJAVE ON PJESMA
    FOR EACH ROW
    BEGIN
    IF :NEW.DATUM_OBJAVE > SYSDATE THEN
        raise_application_error(-20001,'DATUM CHECK ERROR');
    END IF;
END DATUM_CHECK;
/
```

Ovaj okidač se aktivira ako je datum objave pjesme nemoguć, tj. u budućnosti.

ZAKLJUČAK

Ovim projektom izmodeliran je pojednostavljen primjer baze podataka jednog glazbenog streaming servisa. Baze podataka su ključne za isporuku glazbe putem streaming servisa, poboljšavajući naše glazbeno iskustvo i omogućujući raznolike funkcionalnosti.