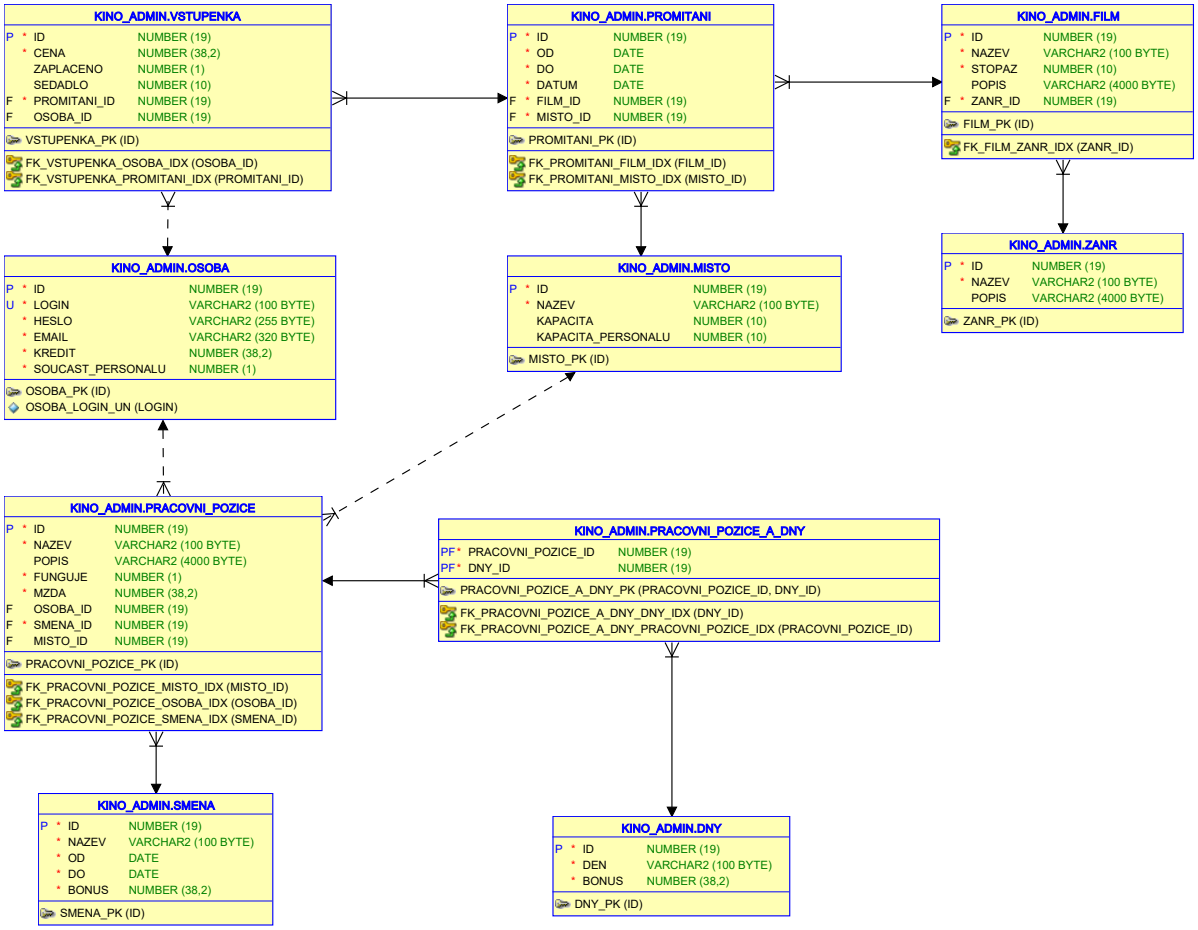
Kino

KIV/DB1 – Semestrální práce

student: Eduard Štich
osobní číslo: A25B0287P
email: stiche@students.zcu.cz
datum: 15. 11. 2025


KINO_ADMIN.VSTUPENKA
P * ID NUMBER (19)
* CENA NUMBER (38,2)
ZAPLACENO NUMBER (1)
SEDADLO NUMBER (10)
F * PROMITANI_ID NUMBER (19)
F OSOBA_ID NUMBER (19)
VSTUPENKA_PK (ID)
FK_VSTUPENKA_OSOBA_IDX (OSOBA_ID)
FK_VSTUPENKA_PROMITANI_IDX (PROMITANI_ID)
KINO_ADMIN.PROMITANI
P * ID NUMBER (19)
* OD DATE
* DO DATE
* DATUM DATE
F * FILM_ID NUMBER (19)
F * MISTO_ID NUMBER (19)
PROMITANI_PK (ID)
FK_PROMITANI_FILM_IDX (FILM_ID)
FK_PROMITANI_MISTO_IDX (MISTO_ID)
KINO_ADMIN.FILM
P * ID NUMBER (19)
* NAZEV VARCHAR2 (100 BYTE)
* STOPAZ NUMBER (10)
POPIS VARCHAR2 (4000 BYTE)
F * ZANR_ID NUMBER (19)
FILM_PK (ID)
FK_FILM_ZANR_IDX (ZANR_ID)
KINO_ADMIN.ZANR
P * ID NUMBER (19)
* NAZEV VARCHAR2 (100 BYTE)
POPIS VARCHAR2 (4000 BYTE)
ZANR_PK (ID)
KINO_ADMIN.OSOBA
P * ID NUMBER (19)
U * LOGIN VARCHAR2 (100 BYTE)
* HESLO VARCHAR2 (255 BYTE)
* EMAIL VARCHAR2 (320 BYTE)
* KREDIT NUMBER (38,2)
* SOUCAST_PERSONALU NUMBER (1)
OSOBA_PK (ID)
OSOBA_LOGIN_UN (LOGIN)
KINO_ADMIN.MISTO
P * ID NUMBER (19)
* NAZEV VARCHAR2 (100 BYTE)
KAPACITA NUMBER (10)
KAPACITA_PERSONALU NUMBER (10)
MISTO_PK (ID)
KINO_ADMIN.PRACOVNI_POZICE
P * ID NUMBER (19)
* NAZEV VARCHAR2 (100 BYTE)
POPIS VARCHAR2 (4000 BYTE)
* FUNGUJE NUMBER (1)
* MZDA NUMBER (38,2)
F OSOBA_ID NUMBER (19)
F * SMENA_ID NUMBER (19)
F MISTO_ID NUMBER (19)
PRACOVNI_POZICE_PK (ID)
FK_PRACOVNI_POZICE_MISTO_IDX (MISTO_ID)
FK_PRACOVNI_POZICE_OSOBA_IDX (OSOBA_ID)
FK_PRACOVNI_POZICE_SMENA_IDX (SMENA_ID)
KINO_ADMIN.PRACOVNI_POZICE_A_DNY
PF * PRACOVNI_POZICE_ID NUMBER (19)
PF * DNY_ID NUMBER (19)
PRACOVNI_POZICE_A_DNY_PK (PRACOVNI_POZICE_ID, DNY_ID)
FK_PRACOVNI_POZICE_A_DNY_DNY_IDX (DNY_ID)
FK_PRACOVNI_POZICE_A_DNY_PRACOVNI_POZICE_IDX (PRACOVNI_POZICE_ID)
KINO_ADMIN.SMENA
P * ID NUMBER (19)
* NAZEV VARCHAR2 (100 BYTE)
* OD DATE
* DO DATE
* BONUS NUMBER (38,2)
SMENA_PK (ID)
KINO_ADMIN.DNY
P * ID NUMBER (19)
* DEN VARCHAR2 (100 BYTE)
* BONUS NUMBER (38,2)
DNY_PK (ID)

1 Detailní popis a charakteristika zadání semestrální práce

1.1 Zdůvodnění výběru tématu

Jako nosné téma databáze jsem zvolil kino, protože jeho provoz tvoří přehlednou síť vzájemně propojených entit (filmy, místa, promítání, vstupenky, osoby), v níž se přirozeně vyskytují všechny požadované konstrukce: vazba M:N, 3. NF a číselníkové tabulky. Díky častým návštěvám různých kin dobře znám některé provozní procesy, a tak dokáži zvolit vhodná testovací data. Toto téma mi zároveň umožní v rámci předmětu KIV/WEB snadno vytvořit webovou aplikaci.

1.2 Vymezení rozsahu

Databáze pokrývá kompletní řetězec „film → promítání → vstupenka → divák" a související agendu se zaměstnanci; nezahrnuje však další provozní okruhy. Uchovává tedy filmy a jejich žánry, místa s kapacitami, časová pásma promítání, jednotlivé vstupenky včetně stavu (vydáno/rezervováno/zaplaceno) a osoby – diváky i zaměstnance – včetně jejich přihlašovacích údajů.

Evidence zahrnuje pracovní pozice, směny, dny a bonusy, avšak neřeší kompletní mzdovou a účetní agendu. Dále neobsahuje skladové hospodaření s občerstvením, dodavatelské faktury, marketingové kampaně ani věrnostní systém; ty jsou ponechány jako možná rozšíření v navazujících předmětech.

1.3 Slovní charakteristika zpracovávaných dat

Datový model pracuje s devíti základními entitami, které pokrývají kompletní provozní agendu kina.

Žánr (`ZANR`) slouží jako číselníková tabulka obsahující název a popis žánru; každý film je přiřazen právě k jednomu žánru.

Film (`FILM`) uchovává název, stopáž v minutách, stručný popis a cizí klíč na žánr. Jedná se o hlavní informační jednotkou, která je dále vázána na konkrétní promítání.

Místo (`MISTO`) reprezentuje promítací sál nebo jiný prostor kina. Má unikátní název, celkovou kapacitu sedadel a současně maximální počet zaměstnanců, kteří se zde mohou teoreticky vyskytovat během směny.

Promítání (`PROMITANI`) spojuje film, sál a časový údaj – slouce s cizími klíči vázané na film a místo, dále sloupec „od" a „do" který určuje čas začátku a konce, sloupec „datum" jako kalendářní den; jedno promítání může mít libovolný počet prodaných vstupenek – teoreticky až tolik, kolik činí kapacita místa.

Osoba (`OSOBA`) eviduje fyzické osoby s unikátním přihlašovací jménem, heslem (v realizované aplikaci pak hashované), kontaktním e-mailem a aktuálním kreditem v Kč. Příznak „soucast_personalu" rozlišuje běžného diváka od zaměstnance.

Vstupenka (`VSTUPENKA`) obsahuje cenu, příznak zaplaceno, číslo sedadla a cizí klíče na promítání a volitelně na osobu; každá vstupenka existuje v jediném exempláři a její stav určuje, zda je rezervace závazná.

Smena (`SMENA`) definuje časový úsek práce („od" a „do") spolu s případným bonusem na hodinu; na jednu směnu je navázána sada pracovních pozic.

Pracovni pozice (`PRACOVNI_POZICE`) konkretizuje roli zaměstnance (např. pokladní, promítač, úklid), popis činnosti, základní mzdu a příznak „funguje" určující, zda je daná pozice aktivní.

Den (`DNY`) představuje číselníkovou tabulku pracovních dnů v týdnu s možností individuálního hodinového bonusu v daný den; přes asociativní tabulku `PRACOVNI_POZICE_A_DNY` je poté možné určit dny ve které se pracovní pozice vykonává.

1.4 Omezení a specifické požadavky

1.4.1 Integritní omezení

`CHECK` „stopaz“ >= 0 zajistí, že délka filmu bude nezáporná.

`CHECK` „kapacita“ >= 0 zajišťuje nezápornou teoretickou kapacitu prostoru pro diváky.

`CHECK` „kapacita_personalu“ >= 0 zajišťuje nezápornou teoretickou kapacitu prostoru pro personál.

`CHECK` „cena“ >= 0 zajišťuje nezápornou cenu vstupenky.

`CHECK` „mzda“ >= 0 zajišťuje nezáporný základ mzdy pro pracovní pozici.

`CHECK` „sedadlo“ > 0 zajišťuje kladné číslo sedadla, jež odpovídá systému kina.

`UNIQUE` „login“ znemožňuje duplicitní přihlašovací jména.

1.4.2 Referenční integrita

Všechny cizí klíče jsou definovány jako `NOT NULL` (kromě „osoba_id“ u vstupenky a pracovní pozice, a „misto_id“ u pracovní pozice), což umožňuje evidovat anonymní či nezakoupenou vstupenku, neobsazenou pozici a pozici s prozatím nenaplánovaným místem výkonu práce.

Mazání nadřazených záznamů je omezeno existujícími vazbami – např. nelze smazat záznam v tabulce `FILM`, na který existuje záznam s cizím klíčem v `PROMITANI`.

1.4.3 Business pravidla implementovaná na aplikační úrovni

Počet vstupenek na promítání by v běžných případech neměl překročit kapacitu místa (`MISTO.kapacita`). Toto pravidlo není vynuceno na úrovni databáze – kontroluje ho až aplikační vrstva.

Počet zaměstanců na určité pracovní pozici, v daný čas, a na daném místě by měl odpovídat kapacitě personálu daného místa (`MISTO.kapacita_personalu`), aby pozice mohla fungovat. Toto omeznení bude kontrolovat až aplikační vrstva.

Promítání filmů by se neměla překrývat – kontrolu časového překryvu (atributy „od“, „do“, „datum“ a „misto_id“) řeší aplikační vrstva.

1.4.4 Číselníkové a asociativní tabulky

`ZANR`, `DNY` a `SMENA` slouží jako číselníkové tabulky – obsahují pouze statické údaje, do kterých běžný provoz pouze čte.

Asociativní tabulka `PRACOVNI_POZICE_A_DNY` umožňuje M:N vazbu mezi pozicí a dny, kdy může být vykonávána – bez dodatečných atributů.

1.4.5 Technická omezení

Všechny časové údaje („od“, „do“ a „datum“) jsou uloženy jako `DATE` bez časové zóny – tato aplikace zajišťuje konzistenci.

Maximální rozsah číselných sloupců (`NUMBER(38,2)`) odpovídá české měnové jednotce (Kč) s danou přesností na haléře.

Tato omezení zajišťují základní konzistenci dat a umožňují bezpečný provoz aplikace nad databází bez nutnosti dodatečných triggerů či procedur.

2 Reprezentativní databázové pohledy

Tato kapitola obsahuje databázové pohledy, které poskytují přehledy a souhrnné informace o aktuálním stavu provozu kina.

2.1 Pohled – naklady-celkove.sql

2.1.1 Slovní popis dotazu

Poskytuje celkový týdenní náklad kina na personál vyjádřený jedinou číselnou hodnotou. Výpočet probíhá ve dvou úrovních: nejprve se pro každého zaměstnance (`SOUCAST_PERSONALU = 1`) vypočítá součet nákladů za všechny jeho směny v týdnu jako součty:

$$\big(\text{mzda na pozici} + \text{bonus za směnu} + \text{bonus za den}\big) \times 7$$

kde číslem 7 je převedena 8hodinová směna včetně 1hodinové přestávky na skutečně odpracovaných 7 hodin. Tyto individuální sumy se pak agregují funkcí `SUM` přes všechny přihlašovací jména (atribut „login“), čímž vznikne jediná hodnota `NAKLADY_NA_TYDEN`. Pohled tak slouží jako rychlý ukazatel celkových osobních nákladů za týden.

2.1.2 SQL kód dotazu

```
CREATE OR REPLACE VIEW NAKLADY_NA_TYDEN AS SELECT SUM(NAKLAD) AS
    NAKLADY_NA_TYDEN
FROM (
    SELECT LOGIN,
      SUM(ZA_SMENU) AS NAKLAD
    FROM (
        SELECT o.LOGIN AS LOGIN,
          ((pp.MZDA + s.BONUS + d.BONUS)*7) AS ZA_SMENU
        FROM PRACOVNI_POZICE pp
        JOIN PRACOVNI_POZICE_A_DNY ppd ON ppd.PRACOVNI_POZICE_ID = pp.ID
        JOIN DNY d ON d.ID = ppd.DNY_ID
        JOIN SMENA s ON s.ID = pp.SMENA_ID
        JOIN OSOBA o ON o.ID = pp.OSOBA_ID
        JOIN MISTO m ON m.ID = pp.MISTO_ID
        WHERE o.SOUCAST_PERSONALU = 1
        ORDER BY pp.NAZEV, o.LOGIN, s.NAZEV, m.NAZEV
    )
    GROUP BY LOGIN
);
```

2.1.3 Odpověď na dotaz

Tabulka 1: Odpověď na naklady-celkove.sql

NAKLADY_NA_TYDEN
88900

2.2 Pohled – naklady-jednotlive.sql

2.2.1 Slovní popis dotazu

Generuje detailní přehled týdenních nákladů rozpočítaný na jednotlivé zaměstnance. Pro každý záznam pracovní pozice a dne se nejprve vypočítá náklad za směnu stejným vzorcem:

$$\big(\text{mzda na pozici} + \text{bonus za směnu} + \text{bonus za den}\big) \times 7$$

Hodnoty se následně seskupí podle `LOGIN` a sečtou, čímž vznikne dvojice `LOGIN` a `NAKLADY_ZA_TYDEN`. Díky tomu pro příklad manažer okamžitě vidí, kolik stojí provoz konkrétního člena personálu za celý týden.

2.2.2 SQL kód dotazu

```
CREATE OR REPLACE VIEW NAKLADY_NA_TYDEN_JEDNOTLIVE AS SELECT LOGIN,
       SUM(ZA_SMENU) AS NAKLADY_ZA_TYDEN
FROM (
    SELECT o.LOGIN AS LOGIN,
           ((pp.MZDA + s.BONUS + d.BONUS)*7) AS ZA_SMENU
    FROM PRACOVNI_POZICE pp
    JOIN PRACOVNI_POZICE_A_DNY ppd ON ppd.PRACOVNI_POZICE_ID = pp.ID
    JOIN DNY d ON d.ID = ppd.DNY_ID
    JOIN SMENA s ON s.ID = pp.SMENA_ID
    JOIN OSOBA o ON o.ID = pp.OSOBA_ID
    JOIN MISTO m ON m.ID = pp.MISTO_ID
    WHERE o.SOUCAST_PERSONALU = 1
    ORDER BY pp.NAZEV, o.LOGIN, s.NAZEV, m.NAZEV
)
GROUP BY LOGIN;
```

2.2.3 Odpověď na dotaz

Tabulka 2: Odpověď na naklady-jednotlive.sql

LOGIN	NAKLADY_ZA_TYDEN
petr	5075
karel	4287,5
lucie	6125
ondra	5162,5
majitel	21000
jana	3762,5
vasek	3412,5
monika	8750
admin	31325

2.3 Pohled – pracovni-pozice-info.sql

2.3.1 Slovní popis dotazu

Podává úplnou mapu obsazených pracovních pozic včetně finančního ohodnocení. Každý řádek odpovídá unikátní kombinaci pozice, zaměstnanec, směna, den a místo. Sloupce `ZA_HODINU` a `ZA_SMENU` počítají přímé osobní náklady podle vzorců:

$$\text{mzda na pozici} + \text{bonus za směnu} + \text{bonus za den}$$

$$(\text{mzda na pozici} + \text{bonus za směnu} + \text{bonus za den}) \times 7$$

Příznak `FUNGUJE` převádí bitovou hodnotu `pp.FUNGUJE` na čitelné „ANO“ / „NE“. Filtr `SOUCAST_PERSONALU = 1` omezuje výstup pouze na aktivní zaměstnance; řazení podle názvu pozice, přihlašovacího jména, směny a místa usnadňuje kontrolu plánu i rozpočtu.

2.3.2 SQL kód dotazu

```
CREATE OR REPLACE VIEW PRACOVNI_POZICE_INFO AS SELECT pp.NAZEV AS
   PRACOVNI_POZICE,
       s.NAZEV AS SMENA,
       o.LOGIN AS ZAMESTNANEC,
       m.NAZEV AS MISTO,
       d.DEN AS DEN,
       (pp.MZDA + s.BONUS + d.BONUS) AS ZA_HODINU,
       ((pp.MZDA + s.BONUS + d.BONUS)*7) AS ZA_SMENU,
       CASE pp.FUNGUJE
            WHEN 1 THEN 'ANO'
            WHEN 0 THEN 'NE'
       END AS FUNGUJE
FROM PRACOVNI_POZICE pp
JOIN PRACOVNI_POZICE_A_DNY ppd ON ppd.PRACOVNI_POZICE_ID = pp.ID
JOIN DNY d ON d.ID = ppd.DNY_ID
JOIN SMENA s ON s.ID = pp.SMENA_ID
JOIN OSOBA o ON o.ID = pp.OSOBA_ID
JOIN MISTO m ON m.ID = pp.MISTO_ID
WHERE o.SOUCAST_PERSONALU = 1
ORDER BY pp.NAZEV, o.LOGIN, s.NAZEV, m.NAZEV;
```

2.3.3 Odpověď na dotaz

Tabulka 3: Odpověď na pracovni-pozice-info.sql (první část)

PRACOVNI_POZICE	SMENA	ZAMESTNANEC	MISTO	DEN	ZA_HODINU	ZA_SMENU	FUNGUJE
IT specialista	Noční	admin	Neurčeno	PO	312,5	2187,5	ANO
IT specialista	Noční	admin	Neurčeno	ÚT	312,5	2187,5	ANO
IT specialista	Noční	admin	Neurčeno	ST	312,5	2187,5	ANO
IT specialista	Noční	admin	Neurčeno	NE	325	2275	ANO
IT specialista	Noční	admin	Neurčeno	ČT	312,5	2187,5	ANO
IT specialista	Noční	admin	Neurčeno	PÁ	337,5	2362,5	ANO
IT specialista	Noční	admin	Neurčeno	SO	325	2275	ANO
IT specialista	Odpolední	admin	Neurčeno	NE	325	2275	ANO
IT specialista	Odpolední	admin	Neurčeno	SO	325	2275	ANO
IT specialista	Odpolední	admin	Neurčeno	PÁ	337,5	2362,5	ANO
IT specialista	Odpolední	admin	Neurčeno	ST	312,5	2187,5	ANO
IT specialista	Odpolední	admin	Neurčeno	ÚT	312,5	2187,5	ANO
IT specialista	Odpolední	admin	Neurčeno	PO	312,5	2187,5	ANO
IT specialista	Odpolední	admin	Neurčeno	ČT	312,5	2187,5	ANO
Obsluha občerstvení	Odpolední	vasek	Občerstvení	ÚT	162,5	1137,5	ANO
Obsluha občerstvení	Odpolední	vasek	Občerstvení	PO	162,5	1137,5	ANO
Obsluha občerstvení	Odpolední	vasek	Občerstvení	ST	162,5	1137,5	ANO
Obsluha projektoru	Odpolední	lucie	Velký sál	ÚT	137,5	962,5	ANO
Obsluha projektoru	Odpolední	lucie	Velký sál	PO	137,5	962,5	ANO
Obsluha projektoru	Odpolední	lucie	Velký sál	NE	150	1050	ANO
Obsluha projektoru	Odpolední	lucie	Velký sál	SO	150	1050	ANO
Obsluha projektoru	Odpolední	lucie	Velký sál	PÁ	162,5	1137,5	ANO
Obsluha projektoru	Odpolední	lucie	Venkovní zátiší	ST	137,5	962,5	ANO
Obsluha projektoru	Odpolední	ondra	Velký sál	PO	137,5	962,5	ANO
Obsluha projektoru	Odpolední	ondra	Velký sál	ÚT	137,5	962,5	ANO
Obsluha projektoru	Odpolední	ondra	Velký sál	PÁ	162,5	1137,5	ANO
Obsluha projektoru	Odpolední	ondra	Velký sál	SO	150	1050	ANO
Obsluha projektoru	Odpolední	ondra	Velký sál	NE	150	1050	ANO

Tabulka 4: Odpověď na pracovni-pozice-info.sql (druhá část)

PRACOVNI_POZICE	SMENA	ZAMESTNANEC	MISTO	DEN	ZA_HODINU	ZA_SMENU	FUNGUJE
Obsluha kiosku	Noční	majitel	Kiosek	SO	175	1225	ANO
Obsluha kiosku	Noční	majitel	Kiosek	PÁ	187,5	1312,5	ANO
Obsluha kiosku	Noční	majitel	Kiosek	ST	162,5	1137,5	ANO
Obsluha kiosku	Noční	majitel	Kiosek	ÚT	162,5	1137,5	ANO
Obsluha kiosku	Noční	majitel	Kiosek	PO	162,5	1137,5	ANO
Obsluha kiosku	Noční	majitel	Kiosek	NE	175	1225	ANO
Obsluha kiosku	Odpolední	majitel	Kiosek	PO	162,5	1137,5	ANO
Obsluha kiosku	Odpolední	majitel	Kiosek	ÚT	162,5	1137,5	ANO
Obsluha kiosku	Odpolední	majitel	Kiosek	ST	162,5	1137,5	ANO
Obsluha kiosku	Odpolední	majitel	Kiosek	PÁ	187,5	1312,5	ANO
Obsluha kiosku	Odpolední	majitel	Kiosek	SO	175	1225	ANO
Obsluha kiosku	Odpolední	majitel	Kiosek	NE	175	1225	ANO
Obsluha kiosku	Ranní	majitel	Kiosek	SO	162,5	1137,5	ANO
Obsluha kiosku	Ranní	majitel	Kiosek	NE	162,5	1137,5	ANO
Obsluha kiosku	Ranní	majitel	Kiosek	ST	150	1050	ANO
Obsluha kiosku	Ranní	majitel	Kiosek	PÁ	175	1225	ANO
Obsluha kiosku	Ranní	majitel	Kiosek	ÚT	150	1050	ANO
Obsluha kiosku	Ranní	majitel	Kiosek	PO	150	1050	ANO
Obsluha občerstvení	Odpolední	jana	Občerstvení	SO	175	1225	ANO
Obsluha občerstvení	Odpolední	jana	Občerstvení	NE	175	1225	ANO
Obsluha občerstvení	Odpolední	jana	Občerstvení	PÁ	187,5	1312,5	ANO
Úklid	Ranní	monika	Veřejné prostory	SO	212,5	1487,5	ANO
Úklid	Ranní	monika	Veřejné prostory	NE	212,5	1487,5	ANO
Úklid	Ranní	monika	Veřejné prostory	PÁ	225	1575	ANO
Úklid	Ranní	monika	Veřejné prostory	ST	200	1400	ANO
Úklid	Ranní	monika	Veřejné prostory	ÚT	200	1400	ANO
Úklid	Ranní	monika	Veřejné prostory	PO	200	1400	ANO
Uvaděč	Odpolední	karel	Velký sál	PÁ	137,5	962,5	ANO
Uvaděč	Odpolední	karel	Velký sál	ÚT	112,5	787,5	ANO
Uvaděč	Odpolední	karel	Velký sál	PO	112,5	787,5	ANO
Uvaděč	Odpolední	karel	Velký sál	NE	125	875	ANO
Uvaděč	Odpolední	karel	Velký sál	SO	125	875	ANO
Uvaděč	Odpolední	petr	Velký sál	SO	125	875	ANO
Uvaděč	Odpolední	petr	Velký sál	NE	125	875	ANO
Uvaděč	Odpolední	petr	Velký sál	PO	112,5	787,5	ANO
Uvaděč	Odpolední	petr	Velký sál	ÚT	112,5	787,5	ANO
Uvaděč	Odpolední	petr	Velký sál	PÁ	137,5	962,5	ANO
Uvaděč	Odpolední	petr	Venkovní zátiší	ST	112,5	787,5	ANO

2.4 Pohled – promitani-info.sql

2.4.1 Slovní popis dotazu

Sestavuje kompletní statistiky každého promítání. Agregací přes `ID` promítání získá jeden řádek obsahující název filmu, formátované datum, název sálu, celkový počet vydaných vstupenek (včetně rezervací), počet skutečně zaplacených vstupenek a kapacitu sálu. Rozdíl mezi kapacitou a vydanými vstupenkami okamžitě prozradí obsazenost; poměr prodaných k vydaným ukazuje míru úspěšného prodeje. Pohled tedy slouží jako operativní dashboard pro kontrolu návštěvnické poptávky.

2.4.2 SQL kód dotazu

```
CREATE OR REPLACE VIEW PROMITANI_INFO AS SELECT f.NAZEV,
       TO_CHAR(p.DATUM, 'DD. MM. YYYY') AS DATUM,
       m.NAZEV AS MISTO,
       COUNT(v.ZAPLACENO) AS VYDANE_VSTUPENKY,
       COUNT(CASE WHEN v.ZAPLACENO = 1 THEN 1 END) AS PRODANE_VSTUPENKY,
       m.KAPACITA
FROM PROMITANI p
JOIN VSTUPENKA v ON v.PROMITANI_ID = p.ID
JOIN FILM f ON f.ID = p.FILM_ID
JOIN MISTO m ON m.ID = p.MISTO_ID
GROUP BY p.ID, f.NAZEV, p.DATUM, m.NAZEV, m.KAPACITA;
```

2.4.3 Odpověď na dotaz

Tabulka 5: Odpověď na promitani-info.sql

NAZEV	DATUM	MISTO	VYDANE_VSTUPENKY	PRODANE_VSTUPENKY	KAPACITA
Leon	05. 01. 2026	Velký sál	20	8	50
Matrix	05. 01. 2026	Velký sál	10	8	20
Pátý element	07. 01. 2026	Venkovní zátiší	10	3	20

2.5 Pohled – smena-info.sql

2.5.1 Slovní popis dotazu

Poskytuje čistý výpis všech směn uložených v tabulce `SMENA`. Vrací název směny, počáteční a koncový čas ve formátu `HH24:MI`. Protože neobsahuje žádné filtry ani agregační funkce, funguje jako referenční seznam pracovních úseků, které jsou dále využívány při plánování personálu i při finančních výpočtech v ostatních pohledech.

2.5.2 SQL kód dotazu

```
CREATE OR REPLACE VIEW SMENA_INFO AS SELECT s.NAZEV AS NAZEV,
       TO_CHAR(s.OD, 'HH24:MI') AS ZACATEK,
       TO_CHAR(s.DO, 'HH24:MI') AS KONEC
FROM SMENA s;
```

2.5.3 Odpověď na doataz

Tabulka 6: Odpověď na smena-info.sql

NAZEV	ZACATEK	KONEC
Ranní	06:00	13:59
Odpolední	14:00	21:59
Noční	22:00	05:59
24-hodinová	00:00	23:59

2.6 Pohled – zisky-celkove.sql

2.6.1 Slovní popis dotazu

Na úrovni celého kina sumuje dva klíčové ekonomické ukazatele:

`PLANOVANE_ZISKY` – potenciální tržbu, kdyby všechna místa byla prodána za průměrnou cenu vstupenky

`DOSAVADNI_ZISKY` – skutečné peněžní toky z již zaplacených vstupenek

Výpočet probíhá nejprve pro jednotlivá promítání:

$$\text{cena vstupenky} \times \text{kapacita místa}$$

$$\text{cena vstupenky} \times \text{počet prodaných vstupenek}$$

a následně se výsledky sečtou přes všechny promítání. Pohled tak dává okamžitou odpověď na otázku, jak daleko je kino od maximálního možného výnosu.

2.6.2 SQL kód dotazu

```
CREATE OR REPLACE VIEW ZISKY AS SELECT SUM(ODHAD_ZA_PROMITANI) AS
    PLANOVANE_ZISKY,
        SUM(ZATIM_ZA_PROMITANI) AS DOSAVADNI_ZISKY
FROM (
        SELECT f.NAZEV,
               TO_CHAR(p.DATUM, 'DD. MM. YYYY') AS DATUM,
               (AVG(v.CENA) * m.KAPACITA) AS ODHAD_ZA_PROMITANI,
               (AVG(v.CENA) * COUNT(CASE WHEN v.ZAPLACENO = 1 THEN 1 END)
                  ) AS ZATIM_ZA_PROMITANI
        FROM PROMITANI p
        JOIN VSTUPENKA v ON v.PROMITANI_ID = p.ID
        JOIN FILM f ON f.ID = p.FILM_ID
        JOIN MISTO m ON m.ID = p.MISTO_ID
        GROUP BY p.ID, f.NAZEV, p.DATUM, m.KAPACITA
);
```

2.6.3 Odpověď na doataz

Tabulka 7: Odpověď na zisky-celkove.sql

PLANOVANE_ZISKY	DOSAVADNI_ZISKY
19200	3780

2.7 Pohled – zisky-jednotlive.sql

2.7.1 Slovní popis dotazu

Detailně rozkládá ekonomiku na úroveň jednotlivých promítání. Pro každý projekční blok vypočítá stejným vzorcem:

$$\text{cena vstupenky} \times \text{kapacita místa}$$

$$\text{cena vstupenky} \times \text{počet prodaných vstupenek}$$

Agregace `GROUP BY p.ID, f.NAZEV, p.DATUM, m.KAPACITA` zajistí, že každý řádek odpovídá právě jedné projekci. Výstup umožňuje sledovat, které konkrétní promítání zaostává či překonává očekávanou tržbu, a tedy kde je prostor pro optimalizaci cen nebo marketingu.

2.7.2 SQL kód dotazu

```
CREATE OR REPLACE VIEW ZISKY_JEDNOTLIVE AS SELECT f.NAZEV,
       TO_CHAR(p.DATUM, 'DD. MM. YYYY') AS DATUM,
       (AVG(v.CENA) * m.KAPACITA) AS ODHAD_ZA_PROMITANI,
       (AVG(v.CENA) * COUNT(CASE WHEN v.ZAPLACENO = 1 THEN 1 END)) AS
          ZATIM_ZA_PROMITANI
FROM PROMITANI p
JOIN VSTUPENKA v ON v.PROMITANI_ID = p.ID
JOIN FILM f ON f.ID = p.FILM_ID
JOIN MISTO m ON m.ID = p.MISTO_ID
GROUP BY p.ID, f.NAZEV, p.DATUM, m.KAPACITA;
```

2.7.3 Odpověď na doataz

Tabulka 8: Odpověď na zisky-jednotlive.sql

NAZEV	DATUM	ODHAD_ZA_PROMITANI	ZATIM_ZA_PROMITANI
Leon	05. 01. 2026	10000	1600
Matrix	05. 01. 2026	3200	1280
Pátý element	07. 01. 2026	6000	900

3 Slovně komentované testovací scénáře

V této kapitole jsou popsány testovací scénáře sloužící k ověření funkčnosti databázového modelu a pohledů v kontextu provozu kina.

3.1 Scénář A

Tento scénář simuluje kompletní interakci nového návštěvníka s (neexistující) webovou aplikací kina. Jeho cílem je ověřit, zda systém správně reaguje na běžný uživatelský průchod: zobrazení volných vstupenek, registraci, nákup, platbu a následné zobrazení vlastních vstupenek. Scénář je implementován v jednom SQL skriptu `a-vse.sql`, který postupně volá dílčí skripty (`a1.sql` až `a4.sql`). Každá část odpovídá jednomu kroku diváka v aplikaci a zároveň ověřuje konzistenci příslušných pohledů.

3.1.1 Část – Divák si prohlíží volná místa

1. Divák hledá volné vstupenky na promítání a zobrazí se mu seznam všech volných míst pro všechna promítání.

```
SELECT f.NAZEV AS FILM,
       z.NAZEV AS ZANR,
       TO_CHAR(p.DATUM, 'DD. MM. YYYY') AS DATUM,
       TO_CHAR(p.OD, 'HH24:MI') AS ZACATEK,
       m.NAZEV AS MISTO,
       CASE
        WHEN v.SEDADLO IS NULL THEN 'Nečíslováno'
        ELSE TO_CHAR(v.SEDADLO)
       END AS CISLO_SEDADLA,
       v.CENA AS CENA
FROM VSTUPENKA v
JOIN PROMITANI p ON p.ID = v.PROMITANI_ID
JOIN MISTO m ON m.ID = p.MISTO_ID
JOIN FILM f ON f.ID = p.FILM_ID
JOIN ZANR z ON z.ID = f.ZANR_ID
WHERE v.OSOBA_ID IS NULL;
```

   `PROMITANI_INFO` zatím neobsahuje žádné změny – počet `PRODANE_VSTUPENKY` odpovídá stavu před nákupem.

   Výsledek dotazu představuje „obrazovku“ s dostupnými sedadly, kterou vidí návštěvník.

3.1.2 Část – Registrace, dobití kreditu a nákup dvou vstupenek

1. Divák vyplní formulář (přihlašovací jméno, e-mail, heslo) a registruje se.
2. Divák přejde na svůj profil a dobije svůj kredit na 400 Kč.
3. Vybere pondělní promítání filmu Leon, označí dvě volná místa a koupí dané vstupenky.

```
INSERT INTO OSOBA (LOGIN, HESLO, EMAIL, KREDIT, SOUCAST_PERSONALU)
VALUES ('pan_leto', 'tajneheslo', 'pan_leto@seznam.cz', 0, 0);
COMMIT;

UPDATE OSOBA
SET KREDIT = KREDIT + 400
WHERE LOGIN LIKE 'pan_leto';
COMMIT;

UPDATE VSTUPENKA
SET OSOBA_ID = (SELECT ID FROM OSOBA WHERE LOGIN LIKE 'pan_leto'),
    ZAPLACENO = 1
WHERE ZAPLACENO = 0
AND OSOBA_ID IS NULL
AND PROMITANI_ID = 1
AND ROWNUM <= 2;
COMMIT;

UPDATE OSOBA
SET KREDIT = KREDIT - (
        (SELECT CENA FROM VSTUPENKA WHERE PROMITANI_ID = 1 AND ROWNUM <=
            1)*2
    )
WHERE LOGIN LIKE 'pan_leto';
COMMIT;
```

`PROMITANI_INFO` – sloupec `PRODANE_VSTUPENKY` se okamžitě zvýší o 2.

`ZISKY` a `ZISKY_JEDNOTLIVE` – hodnota `ZATIM_ZA_PROMITANI` se navýší o cenu dvou vstupenek. Žádné personální pohledy (`PRACOVNI_POZICE_INFO`, `NAKLADY_*`) se nemění – operace se týká pouze návštěvníka.

3.1.3 Část – Návštěvník se přesvědčí, že jsou vstupenky opravdu zakoupil

1. Divák zkontroluje volné vstupenky na promítání a zobrazí se mu seznam všech volných míst pro všechna promítání. Divák očekává, že už neuvidí vstupenky, které zakoupil.

```sql
SELECT f.NAZEV AS FILM,
       z.NAZEV AS ZANR,
       TO_CHAR(p.DATUM, 'DD. MM. YYYY') AS DATUM,
       TO_CHAR(p.OD, 'HH24:MI') AS ZACATEK,
       m.NAZEV AS MISTO,
       CASE
        WHEN v.SEDADLO IS NULL THEN 'Nečíslováno'
        ELSE TO_CHAR(v.SEDADLO)
       END AS CISLO_SEDADLA,
       v.CENA AS CENA
FROM VSTUPENKA v
JOIN PROMITANI p ON p.ID = v.PROMITANI_ID
JOIN MISTO m ON m.ID = p.MISTO_ID
JOIN FILM f ON f.ID = p.FILM_ID
JOIN ZANR z ON z.ID = f.ZANR_ID
WHERE v.OSOBA_ID IS NULL;
```

Volné vstupenky – po opětovném spuštění `a3.sql` (stejný dotaz jako `a1.sql`) chybí právě 2 řádky, které nyní mají nastaveno `OSOBA_ID = <divákovo ID>`.

Výsledek dotazu představuje „obrazovku“ s dostupnými sedadly, kterou vidí návštěvník a která již neobsahuje vstupenky zakoupené divákem.

3.1.4 Část – Divák se přesvědčí, že vstupenky jsou jeho

1. Divák na svém profilu zobrazí své vstupenky a zobrazí se mu seznam všech jeho vstupenek.

```sql
SELECT f.NAZEV AS FILM,
       z.NAZEV AS ZANR,
       TO_CHAR(p.DATUM, 'DD. MM. YYYY') AS DATUM,
       TO_CHAR(p.OD, 'HH24:MI') AS ZACATEK,
       m.NAZEV AS MISTO,
       CASE
        WHEN v.SEDADLO IS NULL THEN 'Nečíslováno'
        ELSE TO_CHAR(v.SEDADLO)
       END AS CISLO_SEDADLA,
       v.CENA AS CENA
FROM VSTUPENKA v
JOIN PROMITANI p ON p.ID = v.PROMITANI_ID
JOIN MISTO m ON m.ID = p.MISTO_ID
JOIN FILM f ON f.ID = p.FILM_ID
JOIN ZANR z ON z.ID = f.ZANR_ID
WHERE v.OSOBA_ID = (SELECT ID FROM OSOBA WHERE LOGIN LIKE 'pan_leto');
```

Výsledek dotazu představuje „obrazovku“ s divákovo vstupenkami – obsahuje přesně 2 řádky s filmem Leon, datem a časem pondělního promítání.

Skript `a-vse.sql` simuluje všechny tyto kroky scénáře v tomto pořadí.

3.2 Scénář B

Tento scénář simuluje interní proces přípravy a následného prodeje vstupenek na vybrané promítání z pohledu pokladního či administrátora kina. Cílem je ověřit, zda systém správně reflektuje vydání nových vstupenek, jejich částečný prodej a okamžité dopady na ekonomické přehledy. Celý průběh je implementován v jednom SQL skriptu `b-vse.sql`, který sekvenčně volá dílčí skripty `b1.sql` až `b5.sql`.

3.2.1 Část – Kontrola stavu před vydáním vstupenek

1. Administrátor zkontroluje vstupenky k promítání číslo 3 a ověří, zda nejsou v systému již vydané.

```
SELECT f.NAZEV,
       TO_CHAR(p.OD, 'HH24:MI') AS ZACATEK_PROMITANI,
       TO_CHAR(p.DO, 'HH24:MI') AS KONEC_PROMITANI,
       TO_CHAR(p.DATUM, 'DD. MM. YYYY') AS DATUM,
       v.SEDADLO,
       v.CENA,
       CASE v.ZAPLACENO
            WHEN 1 THEN 'ANO'
            WHEN 0 THEN 'NE'
       END AS ZAPLACENO,
       o.LOGIN
FROM VSTUPENKA v
JOIN PROMITANI p ON p.ID = v.PROMITANI_ID
JOIN FILM f ON f.ID = p.FILM_ID
LEFT JOIN OSOBA o ON o.ID = v.OSOBA_ID
WHERE p.ID = 3;
```

Výsledek je prázdný – potvrzuje, že před zahájením předprodeje neexistují žádné vstupenky k promítání číslo 3.

Pohled `PROMITANI_INFO` pro dané promítání zobrazuje `VYDANE_VSTUPENKY = 0` a `PRODANE_VSTUPENKY = 0`.

3.2.2 Část – Hromadné vydání vstupenek

1. Administrátor se rozhodne vydat nové vstupenky a systém automaticky vygeneruje 30 číslovaných vstupenek (sedadla 1–30) s cenou 220 Kč dle zadání administrátora.

```
INSERT INTO VSTUPENKA (CENA, ZAPLACENO, SEDADLO, PROMITANI_ID)
VALUES (220, 0, 1, 3);

...

INSERT INTO VSTUPENKA (CENA, ZAPLACENO, SEDADLO, PROMITANI_ID)
VALUES (220, 0, 30, 3);
COMMIT;
```

Pohled `PROMITANI_INFO` se změní – sloupec `VYDANE_VSTUPENKY` naroste na 30.

Ekonomické pohledy `ZISKY` a `ZISKY_JEDNOTLIVE` zvýší hodnotu `ODHAD_ZA_PROMITANI` o

$$30 \times 220.$$

3.2.3 Část – Částečný prodej vstupenek zákazníkům

1. V pokladně přijdou dva zákazníci: první (`ID` 2) zakoupí 5 vstupenek, druhý (`ID` 10) zakoupí 1 vstupenku. Pokladní označí vstupenky jako zaplacené a přiřadí je k účtům zákazníků.

```sql
UPDATE VSTUPENKA
SET ZAPLACENO = 1, OSOBA_ID = 2
WHERE ZAPLACENO = 0 AND PROMITANI_ID = 3
  AND OSOBA_ID IS NULL AND ROWNUM <= 5;
COMMIT;

UPDATE OSOBA
SET KREDIT = KREDIT - (220*5)
WHERE ID = 2;
COMMIT;

UPDATE VSTUPENKA
SET ZAPLACENO = 1, OSOBA_ID = 10
WHERE ZAPLACENO = 0 AND PROMITANI_ID = 3
  AND OSOBA_ID IS NULL AND ROWNUM <= 1;
COMMIT;

UPDATE OSOBA
SET KREDIT = KREDIT - 220
WHERE ID = 10;
COMMIT;
```

Pohled `PROMITANI_INFO` – sloupec `PRODANE_VSTUPENKY` naroste na 6.

Pohled `ZISKY` – hodnota `ZATIM_ZA_PROMITANI` vzroste o

$$6 \times 220$$

Osobní pohledy zůstávají beze změny (operace se týká pouze prodeje, nikoli personálu).

3.2.4 Část – Kontrola stavu po prodeji

1. Administrátor znovu zkontroluje vstupenky k promítání číslo 3 a ověří, že 6 vstupenek je prodaných, zbylých 24 stále čeká na zákazníka.

```sql
SELECT f.NAZEV,
       TO_CHAR(p.OD, 'HH24:MI') AS ZACATEK_PROMITANI,
       TO_CHAR(p.DO, 'HH24:MI') AS KONEC_PROMITANI,
       TO_CHAR(p.DATUM, 'DD. MM. YYYY') AS DATUM,
       v.SEDADLO,
       v.CENA,
       CASE v.ZAPLACENO
            WHEN 1 THEN 'ANO'
            WHEN 0 THEN 'NE'
       END AS ZAPLACENO,
       o.LOGIN
FROM VSTUPENKA v
JOIN PROMITANI p ON p.ID = v.PROMITANI_ID
JOIN FILM f ON f.ID = p.FILM_ID
LEFT JOIN OSOBA o ON o.ID = v.OSOBA_ID
WHERE p.ID = 3;
```

Výsledek obsahuje 30 řádků: 6× `ZAPLACENO = 'ANO'` s přiřazeným loginem,
24× `ZAPLACENO = 'NE'` a prázdný login.

Potvrzuje tak správnost předchozích kroků.

3.2.5 Část – Ekonomické shrnutí promítání

1. Vedoucí kina si nechá zobrazit ekonomické shrnutí – systém ukáže, kolik Kč již bylo utrženo a od kterých zákazníků.

```
SELECT CASE
         WHEN o.LOGIN IS NULL THEN '(Zatím neprodané)'
         ELSE o.LOGIN
       END AS LOGIN,
       SUM(v.CENA) AS ZISK_KINA,
       CASE
         WHEN o.KREDIT IS NULL THEN ' '
         ELSE TO_CHAR (o.KREDIT)
       END AS ZUSTATEK_PO_UTRATE
FROM VSTUPENKA v
JOIN PROMITANI p ON p.ID = v.PROMITANI_ID
LEFT JOIN OSOBA o ON o.ID = v.OSOBA_ID
WHERE p.ID = 3
GROUP BY v.OSOBA_ID, o.LOGIN, o.KREDIT;
```

Výsledek obsahuje dva skutečné řádky (osob s `ID` 2 a 10) s částkami 1100 Kč a 220 Kč a zůstatky kreditů po nákupu.

Jeden řádek `'(Zatím neprodané)'` obsahuje částku 5 280 Kč (24×220) – plánovaný zisk z dosud neprodaných vstupenek.

Skript `b-vse.sql` simuluje všechny tyto kroky scénáře v tomto pořadí.

3.3 Scénář C

Tento scénář demonstruje interní proces změny pracovní pozice zaměstnance v (neexistující) personální aplikaci kina. Jeho cílem je ověřit, zda systém správně zaznamená uvolnění původních pozic a přiřazení nové pozice jedinému zaměstnanci. Celý průběh je implementován v jednom SQL skriptu `c-vse.sql`, který postupně volá skripty `c1.sql` až `c3.sql`.

3.3.1 Část – Zobrazení aktuálně volných pracovních pozic

1. Vedoucí personálního oddělení si zobrazí přehled všech pozic, které aktuálně nemají přiřazeného zaměstnance.

```
SELECT pp.NAZEV AS PRACOVNI_POZICE,
       s.NAZEV AS SMENA,
       CASE
         WHEN pp.OSOBA_ID IS NULL THEN 'Neurčen'
         ELSE TO_CHAR(pp.OSOBA_ID)
       END AS ZAMESTNANEC,
       CASE
         WHEN pp.MISTO_ID IS NULL THEN 'Neurčeno'
         ELSE TO_CHAR(m.NAZEV)
       END AS MISTO
FROM PRACOVNI_POZICE pp
JOIN SMENA s ON s.ID = pp.SMENA_ID
LEFT JOIN MISTO m ON m.ID = pp.MISTO_ID
WHERE pp.OSOBA_ID IS NULL;
```

Výsledek dotazu představuje „obrazovku“ s volnými pracovními pozicemi, které jsou k dispozici pro nové nebo stávající zaměstnance.

Pohled `PRACOVNI_POZICE_INFO` zobrazuje pouze pozice s přiřazenými zaměstnanci, proto volné pozice v tomto pohledu chybí.

3.3.2 Část – Zaměstnanec si vybere novou pozici a uvolní staré

1. Zaměstnanec s přihlašovacím jménem `vasek` projeví zájem o pozici „Noční hlídač“. Personální oddělení mu tuto pozici přiřadí a současně uvolní všechny jeho dosavadní pozice, protože nová role je časově náročná.

```
UPDATE PRACOVNI_POZICE
SET OSOBA_ID = (
    SELECT ID FROM OSOBA WHERE LOGIN LIKE 'vasek'
),
FUNGUJE = 1
WHERE NAZEV LIKE 'Noční hlídač';
COMMIT;

UPDATE PRACOVNI_POZICE
SET OSOBA_ID = NULL,
FUNGUJE = 0
WHERE NAZEV NOT LIKE 'Noční hlídač'
  AND OSOBA_ID = (
      SELECT ID FROM OSOBA WHERE LOGIN LIKE 'vasek'
);
COMMIT;
```

Pohled `PRACOVNI_POZICE_INFO` se změní – zaměstnanec `vasek` je nyní viditelný pouze u pozice „Noční hlídač“.

Pohledy `NAKLADY_NA_TYDEN` a `NAKLADY_NA_TYDEN_JEDNOTLIVE` se automaticky přepočítají.

Ekonomické pohledy `ZISKY` a `ZISKY_JEDNOTLIVE` se změní, protože „Noční hlídač“ je nově vykonávaná a určitá jinačí už nikoliv.

3.3.3 Část – Kontrola aktuálně volných pozic po změně

1. Vedoucí personálního oddělení znovu zobrazí seznam volných pozic a ověří, které pozice byly uvolněny a které nově obsazeny.

```sql
SELECT pp.NAZEV AS PRACOVNI_POZICE,
       s.NAZEV AS SMENA,
       CASE
        WHEN pp.OSOBA_ID IS NULL THEN 'Neurčen'
        ELSE TO_CHAR(pp.OSOBA_ID)
       END AS ZAMESTNANEC,
       CASE
        WHEN pp.MISTO_ID IS NULL THEN 'Neurčeno'
        ELSE TO_CHAR(m.NAZEV)
       END AS MISTO
FROM PRACOVNI_POZICE pp
JOIN SMENA s ON s.ID = pp.SMENA_ID
LEFT JOIN MISTO m ON m.ID = pp.MISTO_ID
WHERE pp.OSOBA_ID IS NULL;
```

Výsledek obsahuje všechny pozice, které byly dříve vykonávané zaměstnancem `vasek`, nyní označené jako 'Neurcen'.

Zároveň chybí pozice „Noční hlídač“, což potvrzuje, že byla úspěšně přiřazena.

Tento krok slouží jako kontrola správnosti předchozích SQL příkazů a zajišťuje konzistenci dat v personálním systému.

Skript `c-vse.sql` simuluje všechny tyto kroky scénáře v tomto pořadí.

Závěr

V rámci semestrální práce se mi podařilo vytvořit komplexní databázový model pro podporu provozu kina, který pokrývá klíčové oblasti – od promítání filmů, prodeje vstupenek až po personální a ekonomické přehledy. Databáze je navržena v 3. normální formě, obsahuje vazby M:N, číselníkové tabulky a implementuje základní integritní a referenční omezení. Díky tomu je datový model konzistentní a připravený pro reálné nasazení, byť asi v omezeném provozu, protože by v produkčním prostředí pravděpodobně vyžadoval rozšíření o další funkce jako např. skladové hospodaření, věrnostní systém nebo účetní agendu.

Během realizace jsem si procvičil práci s relační databází Oracle, psaní komplexních SQL dotazů a vytváření pohledů pro agregaci dat v prostředí SQL Developer. Práce mě bavila zejména proto, že se týkala konkrétního a představitelného tématu. Mohl jsem při ní využít vlastní zkušenosti z návštěv kina a pracovat s reálnými daty o mých oblíbených filmech.