



FAKULTA
APLIKOVANÝCH VĚD
ZÁPADOČESKÉ
UNIVERZITY
V PLZNI

SEMESTRÁLNÍ PRÁCE Z PŘEDMĚTU KIV/DB2

Hledání min

Patrik Harag

harag@students.zcu.cz

(A18N0084P)

27. dubna 2019

Obsah

1	Zadání	2
2	Datová analýza	6
3	Funkční analýza	6
3.1	Trigger T_OBLAST_BEFORE_INSERT	6
3.2	Trigger T_OBLAST_AFTER_INSERT	7
3.3	Procedura ZAMINUJ_OBLAST	8
3.4	Funkce SPATNY_PARAMETR	8
3.5	Pohled VITEZOVE	9
3.6	Pohled PORAZENI	9
4	Scénář hry	9
4.1	Zahájení hry	9
4.2	Označení miny	10
4.3	Odkrytí políčka	10
4.4	Zobrazení herní plochy	10
4.5	Ostatní	11
5	Závěr	11

1 Zadání

Hledání min

Cílem této práce je navrhnout a vytvořit relační databázi pro hraní známé počítačové hry *Hledání min*. Protože bude řešena pouze databázová vrstva aplikace, snažte se co nejvíce programových rutin uložit do databáze a také zajistěte jejich automatickou aktivaci při nastalé události.

Herní oblast má zpravidla tvar obdélníku, ve kterém se nachází několik min. Velikost oblasti a počet min v ní definuje obtížnost hry. Hráč si může vybrat jednu ze tří předdefinovaných obtížností nebo si může definovat obtížnost vlastní. Úkolem hráče je odkrýt všechna pole oblasti, která nejsou zaminována. Hráči se bude od začátku hry měřit čas, aby bylo možné dosažené výsledky porovnávat. Po odkrytí libovolného pole (hráčem nebo databází) může nastat jedna z těchto událostí:

- Hráč šlápl na minu. Hra končí neúspěchem a výsledek se zaznamená do databáze.
- Bylo odkryto poslední pole, na kterém není mina. Hra končí úspěchem, protože zbylá neodkrytá pole obsahují miny. Také v tomto případě se výsledek uloží do databáze.
- Bylo odkryto pole, které je volné a nesousedí s žádným zaminovaným polem. V tomto případě databáze automaticky odkryje všechna sousední pole – ty mají společný minimálně jeden vrchol.

Pro snazší hraní si může hráč označovat ta pole, o kterých si myslí, že jsou zaminovaná. K tomuto rozhodnutí mu pomohou čísla již odkrytých polí, která určují, s kolika zaminovanými poli toto pole sousedí. Takto označené pole nelze odkrýt, ale toto označení lze kdykoliv zrušit.

V relační databázi budou evidována data v těchto tabulkách:

OBTIZNOST	Každá hra musí mít definovanou obtížnost. Tabulka OBLAST bude obsahovat pouze hodnoty parametrů předdefinovaných obtížností. Hodnoty parametrů vlastní obtížnosti se ukládají pouze do tabulky OBLAST. Podle originální hry jsou předdefinované obtížnosti nastaveny takto: <ul style="list-style-type: none">• Začátečník: 9 řádků x 9 sloupců, 10 min• Pokročilý: 16 řádků x 16 sloupců, 40 min• Expert: 16 řádků x 30 sloupců, 99 min
OMEZENI	Každá vlastní obtížnost musí splňovat jistá omezení. Např. počet řádků či sloupců nesmí být menší než 9 a větší než 100. Také je vhodné pohlídat, aby počet rozmístěných min v zaminované oblasti nebyl příliš velký, např. nepřekročil 40 procent její velikosti.
OBLAST	Každá zaminovaná oblast je vytvořená podle předdefinované nebo vlastní obtížnosti a obsahuje její hodnoty. Hráč má za úkol oblast od min vyčistit.
POLE	Elementární část zaminované oblasti definovaná svými souřadnicemi, která může nést minu nebo informaci, s kolika zaminovanými poli sousedí.
MINA	Hráčem označovaná pole, o kterých si myslí, že jsou zaminovaná.
TAH	Hráčem odkrývaná pole v zaminované oblasti. Ke každému tahu se bude automaticky ukládat časová značka, kdy byl tah vykonán.
STAV	Číselník obsahující, v jakých stavech se hra může vyskytovat. Stavy mohou být tyto: <i>rozehraná</i> , <i>úspěšně ukončená</i> nebo <i>neúspěšně ukončená</i> .
HRA	Průběžně aktualizované informace o probíhající hře. Obsahuje časové značky prvního a naposledy provedeného tahu, počet označených min a stav hry.

Z uložených dat v databázi vytvořte databázové pohledy, které nabídnou tato data:

CHYBNE_MINY	Seznam polí v zaminované oblasti, které byly chybně označené jako zaminované. Nabízí data pro všechny oblasti.
VITEZOVE	Výsledková tabulka her, které byly úspěšně dokončené. Měla by ukazovat parametry obtížnosti (rozměry oblasti a počet min) dané hry a také dobu hraní hry (rozdíl časových značek posledního a prvního tahu).
PORAZENI	Výsledková tabulka her, které byly neúspěšně dokončené. Měla by navíc (oproti pohledu VITEZOVE) ukazovat, kolik min bylo správně odhaleno.
OBLAST_TISK	Zobrazení celé zaminované oblasti včetně odkrývaných polí a (hráčem) označených min. Každý řádek oblasti bude zobrazen voláním funkce RADEK_OBLASTI.

V databázi budou uloženy a používány tyto procedury:

ZAMINUJ_OBLAST	Položení min na (náhodná) místa v definované oblasti. Počet zaminovaných polí je uloženo v tabulce OBLAST. Do tabulky POLE ukládá hodnotu -1.
SPOCITEJ_OBLAST	Pro každé nezaminované pole v oblasti spočítá, s kolika zaminovanými poli sousedí. Do tabulky POLE ukládá hodnoty z intervalu 0 až 8.
ODKRYJ_POLE	Rekurzivní procedura, která pro právě odkryté pole, které nesousedí s žádným zaminovaným polem, odkryje všechna jeho neodkrytá sousední pole.
OZNAC_MINY	Po úspěšném dohrání hry budou dosud neodkrytá pole označená jako zaminovaná, tj. vloží se odpovídající záznamy do tabulky MINA.

V databázi budou uloženy a používány tyto funkce:

SPATNY_PARAMETR	Oznámí, zda hodnota parametru vlastní obtížnosti porušila definovaná omezení.
RADEK_OBLASTI	Vrátí řetězec znaků ukazující aktuální podobu daného řádku zaminované oblasti. Pro výpis zvolte vhodné symboly.
ODKRYTA_MINA	Oznámí, že právě odkryté pole skrývá minu, což znamená neúspěšný konec hry.
MNOHO_MIN	Nelze označit více zaminovaných polí, než kolik min je v oblasti.
VYHRA	Počet neodkrytých polí se rovná počtu min, které se v oblasti nachází. Pokud ano, hra končí úspěchem.

O automatické činnosti v databázi se postarají triggerly hlavně o:

- hlídání hodnot parametrů vlastní obtížnosti (volání funkce SPATNY_PARAMETR).
- kopírování hodnot parametrů obtížnosti pro aktuální oblast, pokud byla zvolena základní obtížnost.
- zaminování nastavené oblasti (volání procedury ZAMINUJ_OBLAST) a očíslování polí této oblasti (volání procedury SPOCITEJ_OBLAST).
- volání automatických kontrol (volání funkcí ODKRYTA_MINA a VYHRA) a případně akcí (volání procedury ODKRYJ_POLE) při odkrytí pole.
- zabránění odkrytí již odkrytého pole.
- hlídání počtu polí označených jako zaminované (volání funkce MNOHO_MIN).
- zabránění označení pole jako zaminované, které je již takto označeno nebo je odkryté.
- průběžnou aktualizaci hry po každém jejím tahu.
- označení min, pokud hra skončila úspěšně (volání procedury OZNAC_MINY).
- zabránění odkrytí pole, pokud hra (neúspěšně) skončila.
- zabránění odkrytí pole, které je hráčem označené jako zaminované.

Konfigurace, spuštění a průběh hry

1. Jednorázová konfigurace databáze:
 - a. Naplnění tabulky `OBTIZNOST` daty reprezentující 3 základní obtížnosti hry.
 - b. Naplnění tabulky `OMEZENI` daty definující omezení pro vlastní obtížnost hry.
 - c. Naplnění tabulky `STAV` daty odpovídající různým stavům hry.
2. Hra je zahájena vložением nového záznamu do tabulky `OBLAST`. Automaticky se spustí plnění daty tabulky `POLE`, které reprezentují podobu definované zaminované oblasti. Nakonec se do tabulky `HRA` automaticky vloží nový záznam.
3. Dále má hráč na výběr jednu z možností:
 - a. Zobrazit si aktuální podobu zaminované oblasti prostřednictvím pohledu `OBLAST_TISK`.
 - b. Odkrýt libovolné pole vložением záznamu do tabulky `TAH`. Dále dochází k aktualizaci příslušného záznamu v tabulce `HRA`.
 - c. Označit libovolné pole jako mina vložением záznamu do tabulky `MINA`.
 - d. Zrušit označení zaminovaného pole smazáním odpovídajícího záznamu v tabulce `MINA`.
4. Pokud hra pokračuje, pokračuj bodem 3, jinak bodem 5 (úspěch) nebo 6 (neúspěch).
5. Hra skončila úspěchem. Je vhodné si zobrazit:
 - a. Výsledkovou listinu vítězů voláním pohledu `VITEZOVE`.
 - b. Zobrazení odminované oblasti voláním pohledu `OBLAST_TISK`.
6. Hra skončila neúspěchem. Je vhodné si zobrazit:
 - a. Výsledkovou listinu poražených voláním pohledu `PORAZENI`.
 - b. Seznam chybně označených min voláním pohledu `CHYBNE_MINY`.
7. Novou hru zahájíme bodem 2.


```

-- vlastní obtiznost - nutna kontrola
IF SPATNY_PARAMETR(:new.RADKY, :new.SLOUPCE, :new.MINY) THEN
    raise_application_error(
        -20001, 'Vlastní obtížnost nesplňuje omezení');
END IF;
ELSE
    -- ziskani informaci o obtiznosti
    SELECT * INTO mObtiznost FROM OBTIZNOST
        WHERE ID = :NEW.OBTIZNOST_ID;
    :NEW.RADKY := mObtiznost.RADKY;
    :NEW.SLOUPCE := mObtiznost.SLOUPCE;
    :NEW.MINY := mObtiznost.MINY;
END IF;

-- vyplneni vynechanych hodnot
IF :NEW.ID IS NULL THEN
    :NEW.ID := LAST_ID.NEXTVAL;
END IF;
END;

```

3.2 Trigger T_OBLAST_AFTER_INSERT

Trigger se stará o založení hry, vytvoření polí, zaminování oblasti a vypočtením čísel určujících počet min v okolí, a to po vytvoření nové oblasti.

```

CREATE OR REPLACE TRIGGER T_OBLAST_AFTER_INSERT
AFTER INSERT ON OBLAST FOR EACH ROW
BEGIN
    -- zalozeni hry
    INSERT INTO HRA (ID, OBLAST_ID) VALUES (LAST_ID.NEXTVAL, :new.ID);

    -- vygenerovani poli
    FOR mX IN 1..:NEW.RADKY LOOP
        FOR mY IN 1..:NEW.SLOUPCE LOOP
            INSERT INTO POLE
                (ID, X, Y, MINA, OBLAST_ID)
            VALUES
                (LAST_ID.NEXTVAL, mX, mY, 0, :NEW.ID);
        END LOOP;
    END LOOP;

    -- vygenerovani min
    ZAMINUJ_OBLAST(:NEW.ID, :NEW.RADKY, :NEW.SLOUPCE, :NEW.MINY);
    SPOCITEJ_OBLAST(:NEW.ID);
END;

```


3.3 Procedura ZAMINUJ_OBLAST

Tato procedura slouží k náhodnému zaminování oblasti. Na vstupu má id oblasti, počet řádků, počet sloupců a počet min. Její jádro tvoří cyklus. Při každém opakování je náhodně vygenerována pozice. Pokud na této pozici není mina, dojde k její umístění.

```
CREATE OR REPLACE PROCEDURE ZAMINUJ_OBLAST(
    mId IN INTEGER, mRadky IN INTEGER, mSloupce IN INTEGER,
    mMiny IN INTEGER) AS
    mX INTEGER;
    mY INTEGER;
    mPolozenoMin INTEGER := 0;
    mPoleRow POLE%ROWTYPE;
BEGIN
    WHILE mPolozenoMin < mMiny LOOP
        mX := dbms_random.value(1, mRadky);
        mY := dbms_random.value(1, mSloupce);

        SELECT * INTO mPoleRow
            FROM POLE
            WHERE POLE.OBLAST_ID = mId AND POLE.X = mX AND POLE.Y = mY;

        IF mPoleRow.MINA = 0 THEN
            UPDATE POLE SET POLE.MINA = -1 WHERE POLE.ID = mPoleRow.ID;
            mPolozenoMin := mPolozenoMin + 1;
        END IF;
    END LOOP;
END;
```

3.4 Funkce SPATNY_PARAMETR

Funkce si načte první řádek z tabulky *OMEZENI* a podle něj zvaliduje parametry hry. Pokud jeden z parametrů není validní, vrátí *TRUE*, jinak *FALSE*.

```
CREATE OR REPLACE FUNCTION SPATNY_PARAMETR(mRadky in INTEGER,
    mSloupce in INTEGER, mMiny IN INTEGER) RETURN BOOLEAN AS
    mOmezeni OMEZENI%ROWTYPE;
BEGIN
    SELECT * INTO mOmezeni FROM OMEZENI WHERE ROWNUM = 1;

    IF mRadky NOT BETWEEN mOmezeni.MIN_POLI AND mOmezeni.MAX_POLI THEN
        RETURN TRUE;
    END IF;

    IF mSloupce NOT BETWEEN mOmezeni.MIN_POLI AND mOmezeni.MAX_POLI THEN
        RETURN TRUE;
    END IF;
```

```

    IF (mMiny / (mRadky * mSloupce) * 100) >
        mOmezeni.MAX_POKRYTI_PROCENT THEN
        RETURN TRUE;
    END IF;

    RETURN FALSE;
END;
```

3.5 Pohled VITEZOVE

Zobrazí všechny hry zakončené vítězstvím včetně trvání hry. Provádí se napojení na tabulku *OBLAST*, ze které se zobrazují parametry dané oblasti.

```

CREATE OR REPLACE VIEW VITEZOVE AS
    SELECT h.ID, o.RADKY, o.SLOUPCE, o.MINY,
        (h.POSLEDNI_TAH - h.PRVNI_TAH) AS TRVANI
    FROM HRA h
    INNER JOIN OBLAST o ON h.OBLAST_ID = o.ID
    WHERE h.STAV_ID = 2
    ORDER BY (h.POSLEDNI_TAH - h.PRVNI_TAH);
```

3.6 Pohled PORAZENI

Podobně jako pohled *VITEZOVE* zobrazí všechny hry zakončené porážkou včetně trvání hry. Provádí se napojení na tabulku *OBLAST*, ze které se zobrazují parametry dané oblasti. Navíc obsahuje počet min, které byly správně odhaleny – kvůli tomu obsahuje další vnořený *SELECT*.

```

CREATE OR REPLACE VIEW PORAZENI AS
    SELECT h.ID, o.RADKY, o.SLOUPCE, o.MINY,
        (SELECT COUNT(*) FROM MINA m
            INNER JOIN POLE p ON m.POLE_ID = p.ID
            WHERE p.OBLAST_ID = h.OBLAST_ID AND p.MINA = -1)
        AS MINY_SPRAVNE,
        (h.POSLEDNI_TAH - h.PRVNI_TAH) AS TRVANI
    FROM HRA h INNER JOIN OBLAST o ON h.OBLAST_ID = o.ID
    WHERE h.STAV_ID = 3
    ORDER BY (h.POSLEDNI_TAH - h.PRVNI_TAH);
```

4 Scénář hry

4.1 Zahájení hry

Pro zahájení hry je potřeba nejdříve vložit záznam do tabulky *OBLAST*. Je možné využít buď předpřipravené obtížnosti nebo definovat vlastní. Oba způsoby jsou ukázány níže.

```

INSERT INTO OBLAST
  (OBTIZNOST_ID)
VALUES
  ((SELECT ID FROM OBTIZNOST WHERE NAZEV = 'začátečník'));

```

```

INSERT INTO OBLAST
  (RADKY, SLOUPCE, MINY)
VALUES
  (10, 10, 5);

```

První vytvořená oblast bude mít ID=0. V dalších ukázkách předpokládáme, že to tak je.

4.2 Označení miny

Níže je předvedeno označení miny a zrušení označení.

```

-- označení miny
INSERT INTO MINA
  (POLE_ID)
VALUES
  ((SELECT ID FROM POLE WHERE X = 5 AND Y = 4 AND OBLAST_ID = 0));

-- zrušení označení
DELETE FROM MINA WHERE POLE_ID =
  ((SELECT ID FROM POLE WHERE X = 2 AND Y = 2 AND OBLAST_ID = 0));

-- označení všech min - pro testování
EXECUTE OZNAC_MINY(0);

```

4.3 Odkrytí políčka

Níže je předvedeno odkrytí políčka.

```

-- odkrytí políčka
INSERT INTO TAH
  (POLE_ID)
VALUES
  ((SELECT ID FROM POLE WHERE X = 3 AND Y = 5 AND OBLAST_ID = 0));

-- odkrytí všech nezaminovaných polí - pro testování
EXECUTE ODKRYJ_OBLAST(0);

```

4.4 Zobrazení herní plochy

Níže uvedeným příkazem je možné nechat vypsat herní plochu.

```

SELECT RADKA FROM OBLAST_TISK WHERE OBLAST_ID = 0;

```

Význam znaků:

- Tečka (.) – neodkryté políčko,
- Číslo (0-8) – počet min v okolí u odkrytého políčka,
- Křížek (#) – odkrytá mina (v případě prohry),
- Vykřičník (!) – hráčem označená mina.

4.5 Ostatní

```
-- počet chybně označených min  
SELECT X, Y FROM CHYBNE_MINY WHERE OBLAST_ID = 0;
```

```
-- přehledy  
SELECT * FROM HRA;  
SELECT * FROM VITEZOVE;  
SELECT * FROM PORAZENI;
```

5 Závěr

Práce splňuje zadanou funkcionalitu v plné výši. Navíc byla vytvořena procedura *ODKRYJ_OBLAST* pro testování hry.

Přílohy

- *init.sql* – script inicializující hru (vytvoří tabulky, trigger, funkce...). Po jeho vykonání je možné zahájit scénář hry. V případě opětovného spuštění dojde ke smazání dat a opětovné inicializaci.
- *test.sql* – obsahuje vzorové příkazy scénáře hry.