ADMINISTRACE UŽIVATELŮ

UŽIVATELÉ

- Každý databázový účet má:
  - Unikátní uživatelské jméno
  - Autentifikační metodu
  - Default tablespace
  - Temporary tablespace
  - Uživatelský profil
  - Status uživatelského účtu
  - Kvóty uživatele
- Schéma:
  - Je souhrn všech databázových objektů, které vlastní jeden databázový uživatel
  - Má shodné jméno jako uživatelský účet
  - Může (a dost často bývá) prázdné

PŘEDDEFINOVANÉ ADMINISTRÁTORSKÉ ÚČTY

- SYS:
  - Má DBA roli a několik dalších rolí
  - Má všechna práva s ADMIN OPTION
  - Je potřeba pro startup, shutdown a některé další administrativní úkony
  - Je vlastníkem data dictionary a Automatic Workload Repository (AWR)
- Uživatel SYSTEM
  - má DBA, EXP_FULL_DATABASE a AQ_ADMINISTRATOR_ROLE role
- Uživatel DBSNMP
  - má OEM_MONITOR a CDB_DBA roli.
- Uživatel SYSBACKUP
  - má SELECT_CATALOG_ROLE role
- Tyto účty jsou speciální - administrátorské, nejsou určeny pro běžnou práci
- Od 12c ještě uživatelé SYSASM, SYSDG a SYSKM pro administraci ASM, Data Guardu a TDE (Transparent Data Encryption)

ZALOŽENÍ UŽIVATELE

- Nejjednodušší syntaxe

```
SQL> Create user pepa identified by silneheslo;
```

- Heslo je v tomto případě case-insensitive, pokud chceme case-sensitive, pak ho dáme do uvozovek
- Hesla až do verze 10g byla case-insensitive
- Delší syntaxe:

```
SQL> Create user uzivatel profile mujprofil identified by
    heslo default tablespace mojetablespace temporary
    tablespace temp;
```

AUTENTIFIKACE UŽIVATELŮ

- Heslo – nevyžaduje žádné další nastavení
- External – autentifikace pomocí operačního systému, jiná autentifikační autorita (např. Kerberos, RADIUS), tzv. adresářové služby (LDAP, Active Directory) - musí se nastavit napojení na tuto autoritu, spadá sem i autentizace pomocí certifikátů

AUTENTIFIKACE ADMINISTRÁTORŮ

- Operační systém:
  - Administrátor (DBA) musí mít v operačním systému práva vytvářet a mazat soubory
  - Běžní uživatelé by neměli mít na úrovni operačního systému práva vytvářet nebo mazat soubory
- Administrátoři:
  - `SYSDBA` připojení:
    - Přihlášení buď pomocí OS autentifikace (uživatel oracle na Linuxu, administrator na Win) nebo pomocí password souboru
    - Toto přihlášení je vždy auditováno
    - Autentifikace pomocí OS má přednost před password souborem
    - Password soubor používá case-sensitive hesla
    - K vytváření password souboru používáme utilitu orapwd

ZAMYKÁNÍ ÚČTŮ

- Zamykání a odemykání účtu

```
SQL> Alter user pepa account lock;
```

```
SQL> Alter user pepa account unlock;
```

- Pokud je účet zamčen, uživatel se nepřihlásí do DB
  - Nejrychlejší způsob řešení bezpečnostních problémů a incidentů

PRÁVA

- V Oracle jsou dva typy uživatelských práv:
  - Systémová: Vážou se ke konkrétním úkonům v databázi (např. create table …)
  - Objektová: Práva k jednotlivým objektům, které nevlastním (např. select on table xy)

SYSTÉMOVÁ PRÁVA

- Udělují se příkazem `grant`

```
SQL> Grant create user to pepa;
```

- Odebírají se příkazem `revoke`

```
SQL> Revoke create user from pepa;
```

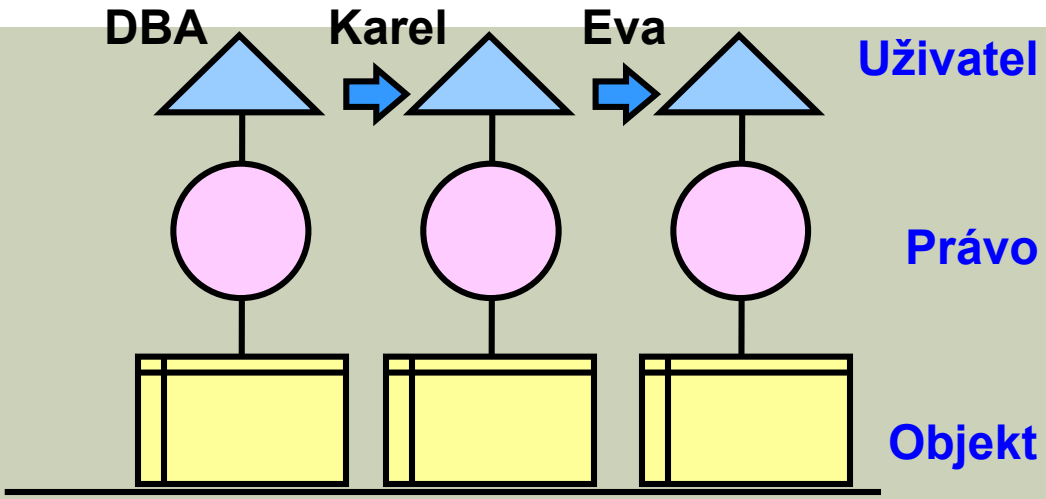
- Základní členění:
  - Zakládání objektů – create user, table, procedure, …
  - Modifikace objektů – alter user, table, procedure, …
  - Rušení objektů – drop user, table, procedure …
- U lokálních objektů právo na založení, modifikaci a rušení vlastních nebo libovolných – create table x create any table
- Další speciální – alter database, alter system, …
- Kdo se chce přihlásit do databáze musí mít právo create session

OBJEKTOVÁ PRÁVA

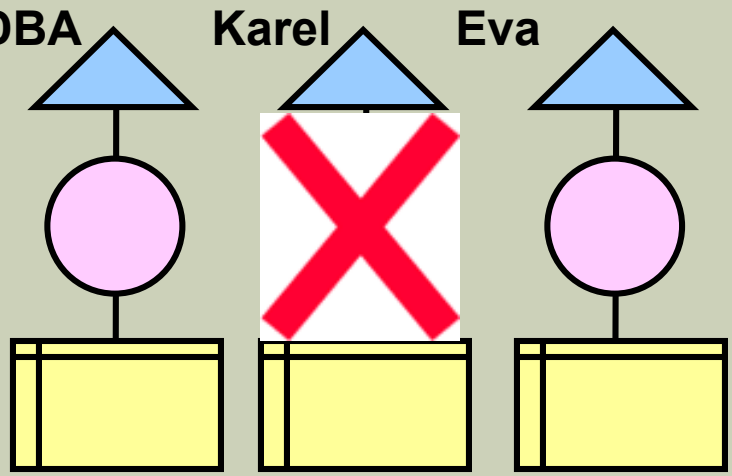
- Pro tabulku 7 základních:
  - select, update, insert, delete
  - alter – změna struktury
  - reference – cizí klíč
  - index
- Pro procedury a funkce
  - execute
- Pro adresář
  - read, write, execute

KLAUZULE WITH ADMIN OPTION

GRANT

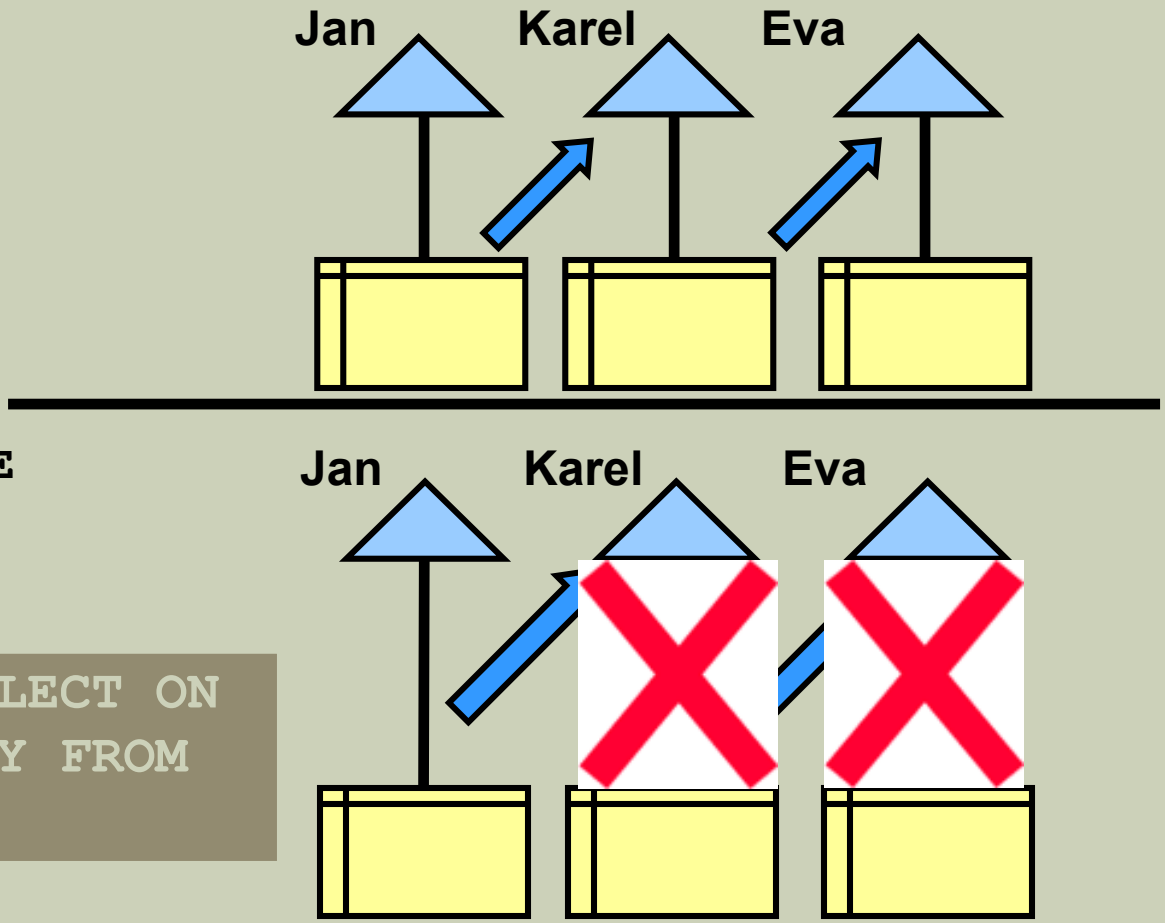


REVOKE




```
REVOKE CREATE
TABLE FROM karel
```

KALUZULE WITH GRANT OPTION




```
REVOKE SELECT ON
TABULKA XY FROM
karel
```

ROLE

- Jednotlivá práva seskupujeme do větších celků a to zejména z důvodů:
  - Jednodušší administrace
  - Umožňují dynamické přidávání nebo odebírání práv
  - Lze je jednoduše vypnout/zapnout (např. při upgrade aplikace)

PŘIŘAZOVÁNÍ ROLÍ A PRÁV

- Lze kombinovat systémová i objektová práva, např. role DEVELOPER může mít práva create table, create function, create procedure a select on ciselniky.cis_statu a select on ciselniky.cis_obci
- Aplikační role – vždy dáváme jen taková práva, která uživatele bezprostředně potřebuje, např. role UCITEL select, update on ZNAMKY, ale už ne insert a delete
- Role může obsahovat jinou roli
- Roli nikdo nevlastní

PŘEDDEFINOVANÉ ROLE

- CONNECT
  - CREATE SESSION, SET CONTAINER
- RESOURCE
  - CREATE CLUSTER, CREATE INDEXTYPE, CREATE OPERATOR, CREATE PROCEDURE, CREATE SEQUENCE, CREATE TABLE, CREATE TRIGGER, CREATE TYPE
- SCHEDULER_ ADMIN
  - CREATE ANY JOB, CREATE EXTERNAL JOB, CREATE JOB, EXECUTE ANY CLASS, EXECUTE ANY PROGRAM, MANAGE SCHEDULER
- DBA
  - Většina systémových práv, některé další role – není určena pro běžné uživatele
- Plus cca 80 dalších …

VYTVOŘENÍ A PŘIŘAZENÍ ROLE

- **Základní syntaxe:**

```
Create role moje_role
```

- **Uživatel může mít jednu nebo více rolí a jednu nebo více defaultních rolí:**

```
Grant moje_role, tvoje_role to uzivatel
Alter user uzivatel default role moje_role
```

- **Pokud není role defaultní, nemůžeme použít žádné z jejích práv – nejprve si ji musíme nastavit:**

```
Set role tvoje_role
```

ZABEZPEČENÍ ROLÍ

- Role mohou být zabezpečeny heslem, je třeba nadefinovat při jejím vytváření:

```
SQL>Create role bezpecna_role identified by
    velmi_bezpecne_heslo;
```

- Při přiřazení pak musíme toto heslo zadat

```
SQL>Set role bezpecna_role identified by bezpecne_heslo;
```

- Další možností je identifikace pomocí procedury

```
SQL>Create role bezpecna_role identified using
    bezpecna_procedura;
```

- Kontrolujeme např. zda je uživatel přihlášen z požadovaného rozsahu adres (např. zevnitř firmy) nebo třeba čas

PROFILY A UŽIVATELÉ

- Každý uživatel má přiřazen právě jeden profil
- Standardní profil je pojmenován DEFAULT
- Profily:
  - Řídí využití zdrojů
    - SESSIONS_PER_USER, CPU_PER_SESSION, CPU_PER_CALL, CONNECT_TIME, IDLE_TIME, LOGICAL_READS_PER_SESSION, LOGICAL_READS_PER_CALL, PRIVATE_SGA, COMPOSITE_LIMIT
  - Nastavují parametry hesla
    - FAILED_LOGIN_ATTEMPTS, PASSWORD_LIFE_TIME, PASSWORD_REUSE_TIME, PASSWORD_REUSE_MAX, PASSWORD_LOCK_TIME, PASSWORD_GRACE_TIME, PASSWORD_VERIFY_FUNCTION
- Inicializační parametr `RESOURCE_LIMIT` musí být nastaven na `TRUE`, pokud chceme profily používat k řízení zdrojů

PROFILY A ŘÍZENÍ ZDROJŮ

- **SESSIONS_PER_USER**
  - kolikrát může být uživatel současně přihlášen
- **CPU_PER_SESSION**
  - kolik CPU může uživatel spotřebovat v setinách sekundy
- **CPU_PER_CALL**
  - kolik CPU může uživatel spotřebovat na jeden příkaz v setinách sekundy
- **CONNECT_TIME**
  - jak dlouho může být uživatel maximálně přihlášen v minutách
- **IDLE_TIME**
  - jak dlouho může být uživatel neaktivní v minutách
- **LOGICAL_READS_PER_SESSION**
  - kolik datových bloků může uživatel maximálně přečíst
- **LOGICAL_READS_PER_CALL**
  - kolik datových bloků může uživatel maximálně přečíst na jeden příkaz
- **PRIVATE_SGA**
  - kolik maximálně SGA může uživatel alokovat
- **COMPOSITE_LIMIT**
  - kombinace CPU_PER_SESSION, CONNECT_TIME, LOGICAL_READS_PER_SESSION, a PRIVATE_SGA pro jednotlivý SQL příkaz – hodnota je tzv. **cost**
- **INACTIVE_ACCOUNT_TIME**
  - pokud se uživatel x dní nepřihlásí, účet se zablokuje

PROFILY A HESLO

- **FAILED_LOGIN_ATTEMPTS**
  - počet neúspěšných přihlášení, po kterých je účet zablokován
- **PASSWORD_LIFE_TIME**
  - počet dní, po němž je nutné změnit heslo
- **PASSWORD_REUSE_TIME a PASSWORD_REUSE_MAX**
  - první určuje za jak dlouho může být použito znovu stejné heslo, druhý pak, kolikrát se musí heslo změnit, než je možné znovu použít stejné – může být nastaveny oba najednou
- **PASSWORD_LOCK_TIME**
  - jak dlouho bude účet zamčen po neúspěšných přihlášeních
- **PASSWORD_GRACE_TIME**
  - pokud heslo vyprší, jak dlouho má uživatel na to, aby si heslo změnil, než se mu účet zablokuje
- **PASSWORD_VERIFY_FUNCTION**
  - definuje funkci, která zkontroluje, zda je heslo dostatečně bezpečné (může kontrolovat např. délku, skupiny znaků …)

VYTVOŘENÍ PROFILE

```
SQL> CREATE PROFILE new_profile
LIMIT
PASSWORD_REUSE_MAX 10
PASSWORD_REUSE_TIME 30;
```

```
SQL> CREATE PROFILE app_user
LIMIT
SESSIONS_PER_USER UNLIMITED
CPU_PER_SESSION UNLIMITED
CPU_PER_CALL 3000
CONNECT_TIME 45
LOGICAL_READS_PER_SESSION DEFAULT LOGICAL_READS_PER_CALL
1000
PRIVATE_SGA 15K
COMPOSITE_LIMIT 5000000;
```

PŘÍKLAD FUNKCE PRO KONTROLU HESLA

- Po instalaci jsou dostupné 3 skripty s příkladem funkce pro kontrolu hesla, základní je `oraxxx_verify_function`, funkce kontroluje:
  - Minimum 8 znaků
  - Rozdílnost oproti uživatelskému jménu, uživatelskému jménu plus číslo a obrácenému uživatelskému jménu
  - Rozdílnost oproti názvu databáze a názvu databáze plus číslice
  - Slovníková kontrola na nejběžnější hesla
  - Alespoň jedno písmenko a alespoň jedna číslice
  - Rozdílnost oproti předchozímu heslu alespoň ve třech znacích
- Defaultní nastavení je – nic se nekontroluje!
- Předdefinované jsou i další funkce pro přísnější kontroly

PŘIHLAŠOVÁNÍ UŽIVATELŮ

- Kromě PROFILE řídí přístup do DB ještě následující inicializační parametry:
  - SEC_MAX_FAILED_LOGIN_ATTEMPTS - default=3
  - SEC_PROTOCOL_ERROR_FURTHER_ACTION - default je drop, 3
  - SEC_PROTOCOL_ERROR_TRACE_ACTION - default je TRACE
  - SEC_RETURN_SERVER_RELEASE_BANNER - default je FALSE
- Kontrola probíhá na úrovni sítě, tj. dojde k odpojení TCP připojení

KVÓTA

- Váže se na TABLESPACE
- Váže se uživatele, resp. jeho objekty, pokud dám právo vkládat data do mých tabulek jiným uživatelů, počítají se i tato data do kvóty
- Systémové právo `UNLIMITED TABLESPACE` umožní nekontrolovat žádné kvóty pro daného uživatele
- Kvóta může být:
  - Číslo a jednotky (K,M nebo G)
  - Unlimited

PRINCIP MINIMA PRÁV

- Aby byla databáze co nejvíce zabezpečena, měli by mít všichni uživatelé jen ta práva, která opravdu bezprostředně potřebují
- Zkontrolujte tedy vždy:
  - Práva k data dictionary – inicializační parametr (default) `O7_DICTIONARY_ACCESSIBILITY=FALSE`
  - Role PUBLIC nemá nejlépe žádná práva
  - Pro přístup do sítě se používá access control lists (ACL)
  - Práva na adresáře nemá nikdo, kdo je nepotřebuje
  - Nikdo kromě uživatelů SYS a SYSTEM nemá administrátorská nebo systémová práva
  - Je nastaven parametr pro znemožnění vzdálené autorizace `REMOTE_OS_AUTHENT=FALSE`

ADMINISTRÁTORSKÉ ÚCTY A ROLE

- Administrátorské účty by vždy měly mít:
  - Silná case-sensitive hesla nebo jiný druh silné autentizace
  - Případné role a administrátorskými právy (např. DBA) by neměly být bez hesla

DOTAZY?