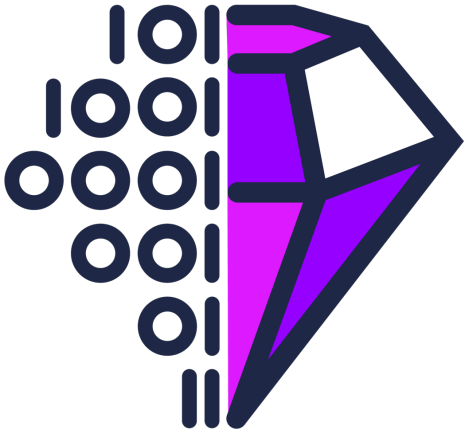
DATENBANK ARCHITEKTUR GRUNDLAGEN

Markus Flechtner

trivadis
Part of Accenture

2 WIR SIND …



- eine international tätige Firma mit Schweizer Wurzeln, die:
  - spezialisiert ist auf **Data Management** und **Data Analysis**, **KI** und **Machine Learning**
  - mit Kunden und Partnern gemeinsam die **digitale Zukunft gestaltet**, indem wir aus Daten Chancen machen
  - **720 Mitarbeitende** an **15 Workspaces** in 4 Ländern beschäftigt

3 DATEN SIND UNSERE DNA

Seit 1994 unterstützen wir Unternehmen dabei, **Daten und neue Technologien intelligent zu nutzen**.

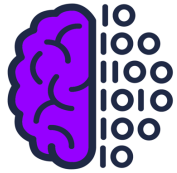
Dabei decken wir das gesamte Spektrum ab: von der **Entwicklung und dem Betrieb von Datenplattformen und Lösungen**, der **Veredelung von Daten** bis hin zur **Beratung** und zum **Training**.

UNSERE DATA VALUE CHAIN

1994

GRÜNDUNG

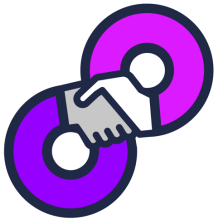
>2200



TRAININGSTEILNEHMENDE PRO JAHR

2100

AUFTRÄGE PRO JAHR



>750

KUNDEN PRO JAHR

123 Mio.

UMSATZ

>300

AKTIVE SLAs

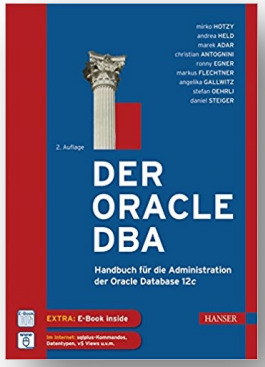
6 UNSERE WORKSPACES

HI!

ORACLE ACE

MARKUS FLECHTNER

PRINCIPAL CONSULTANT



DOAG

- Trivadis Germany GmbH
- Diplom-Mathematiker
- Schwerpunkte
  - Oracle Hochverfügbarkeit
  - Database Upgrade + Migration
- Kursreferent: RAC, New Features, Multitenant, PostgreSQL
- Twitter @markusdba
- Blog: markusdba.net|.de
- Co-Autor des Buches "The Oracle DBA" (2016)

trivadis Part of Accenture

9 AGENDA

Das ist ein Einsteiger-Vortrag –
mit Verkürzungen, Vereinfachungen, etc.

- Grundbegriffe
- Datenbank-Architektur bei Oracle
  - Dateien
  - Prozesse
  - Speicherstrukturen
  - Data Dictionary
- Was passiert bei einer Transaktion?
- Container-Datenbanken bei Oracle
- Links

GRUNDBEGRIFFE

11 GRUNDBEGRIFFE

- Datenbank
- Transaktion
- ACID
- MVCC

12 DATENBANK

- Die Wikipedia sagt:

  "Die wesentliche Aufgabe einer Datenbank ist es,
  große Datenmengen effizient, widerspruchsfrei und dauerhaft zu speichern

  und

  benötigte Teilmengen in unterschiedlichen, bedarfsgerechten Darstellungsformen für Benutzer und Anwendungsprogramme bereitzustellen."

13 RDBMS

- RDBMS steht für "**Relational Database Management System**"
- Daten werden Tabellen gespeichert
  - In der DB-Theorie heißen "Tabellen" auch "Relationen"
- Beispiele
  - Oracle :)
  - Microsoft SQL Server
  - MariaDB / MySQL
  - PostgreSQL

14 TRANSAKTION

- Eine Transaktion ist "eine logische zusammenhängende Operation, die aus mehreren Befehlen bestehen kann"

- SELECT
- INSERT
- UPDATE
- DELETE

15 TRANSAKTION – DAS KLASSISCHE BEISPIEL (ÜBERWEISUNG)

```
SQL> UPDATE konten
     SET konto_stand=konto_stand - 100
     WHERE konto_nummer=4711;
1 row updated.


SQL> UPDATE konten
     SET konto_stand=konto_stand + 100
     WHERE konto_nummer=4712;
1 row updated.


SQL> COMMIT;
Commit complete.
```

16 ACID

- ACID = Anforderungen an Transaktionen in einem Datenbank-Management-System (DBMS)

- "ACID" steht für
  - Atomicity = Eine Transaktion wird entweder ganz oder gar nicht ausgeführt ("either all-or-nothing")
  - Consistency = Nach einer Transaktion sind die Daten logisch konsistent.
  - Isolation = Transaktionen dürfen sich nicht gegenseitig beeinflussen
  - Durability = Auch Systemfehler verändern die Daten nicht.

17 MVCC = MULTI-VERSION-CONCURRENCY-CONTROL

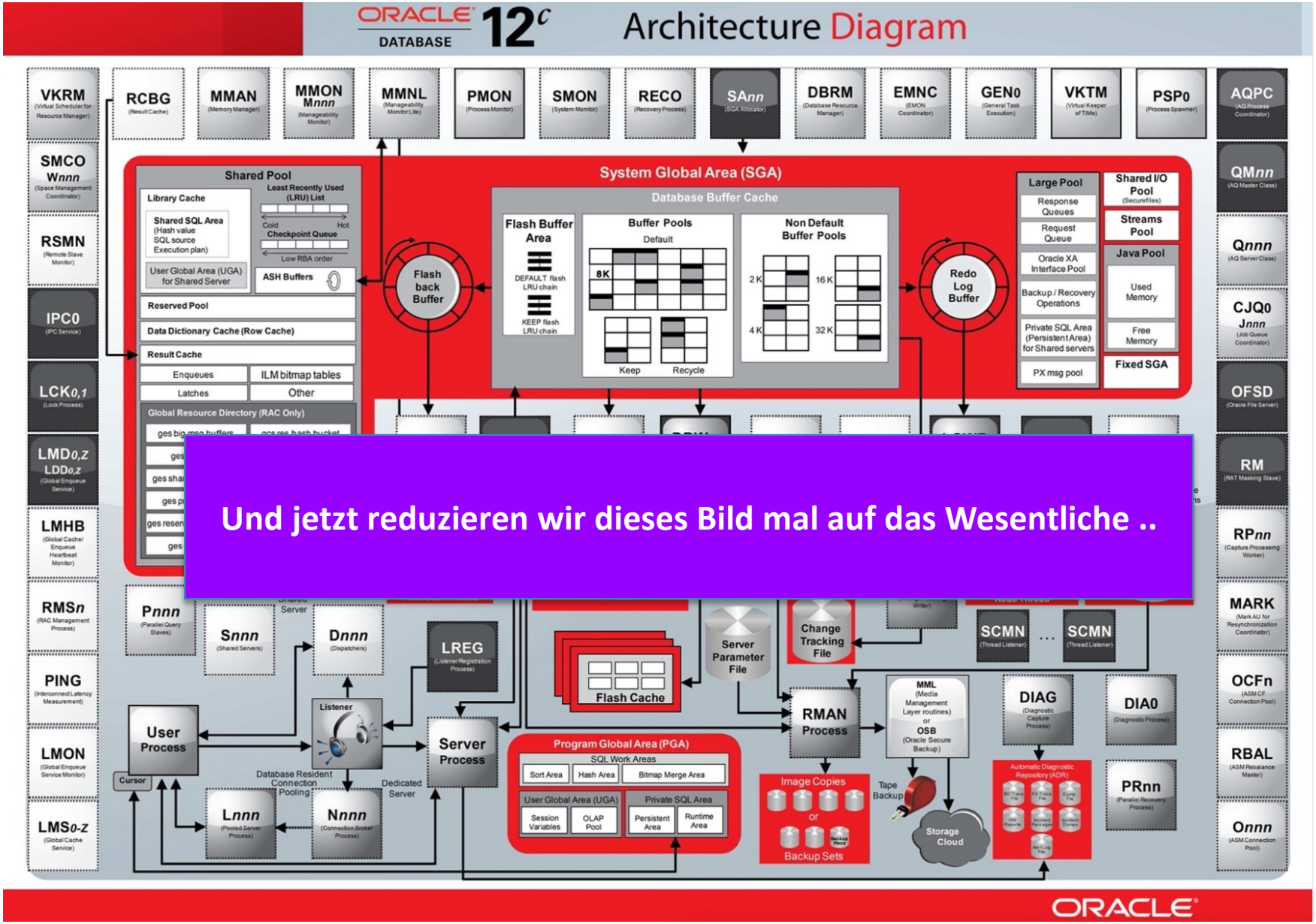
- Konkurriende Zugriffe in einer Datenbank sollen ausgeführt werden können, ohne sich gegenseitig zu blockieren oder die Konsistenz der Datenbank zu gefährden

- "readers don't bl Transaktion sieht nur das, was zu Beginn einer Transaktion commited war

18 IMPLEMENTIERUNG

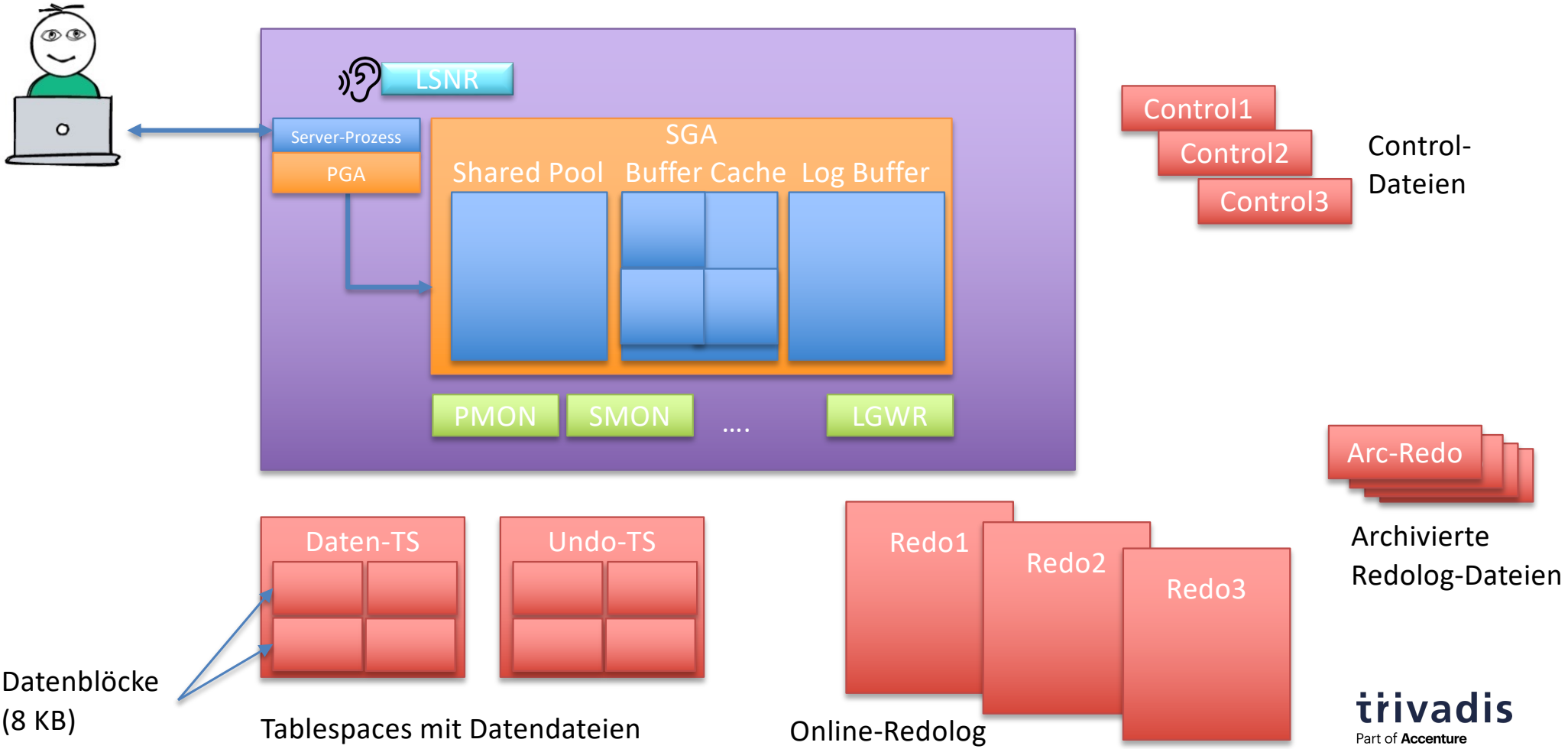
- Alle gängigen aktuellen RDBMS implementieren diese Prinzipien (ACID, MVCC).
- Die Mechanismen sind ähnlich.
- Die Unterschiede liegen im Detail und in den Namen für die jeweiligen Komponenten.

DATENBANK-ARCHITEKTUR ORACLE



trivadis
Part of Accenture

21 THE BIG PICTURE (IMMER NOCH KOMPLIZIERT GENUG)

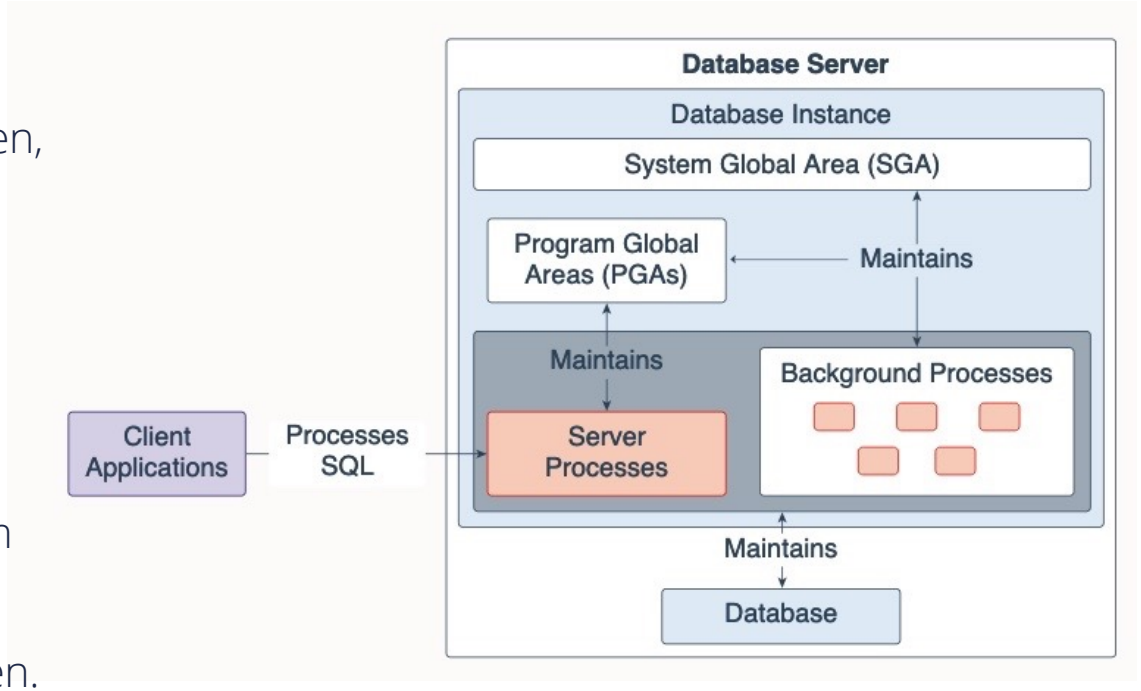


trivadis
Part of Accenture

22 DATENBANK & INSTANZ

- Datenbank
  - Eine Datenbank ist eine Menge von Dateien, die sich auf der Festplatte befinden und Daten speichern.
  - Datenbankdateien können ohne eine Instanz existieren.

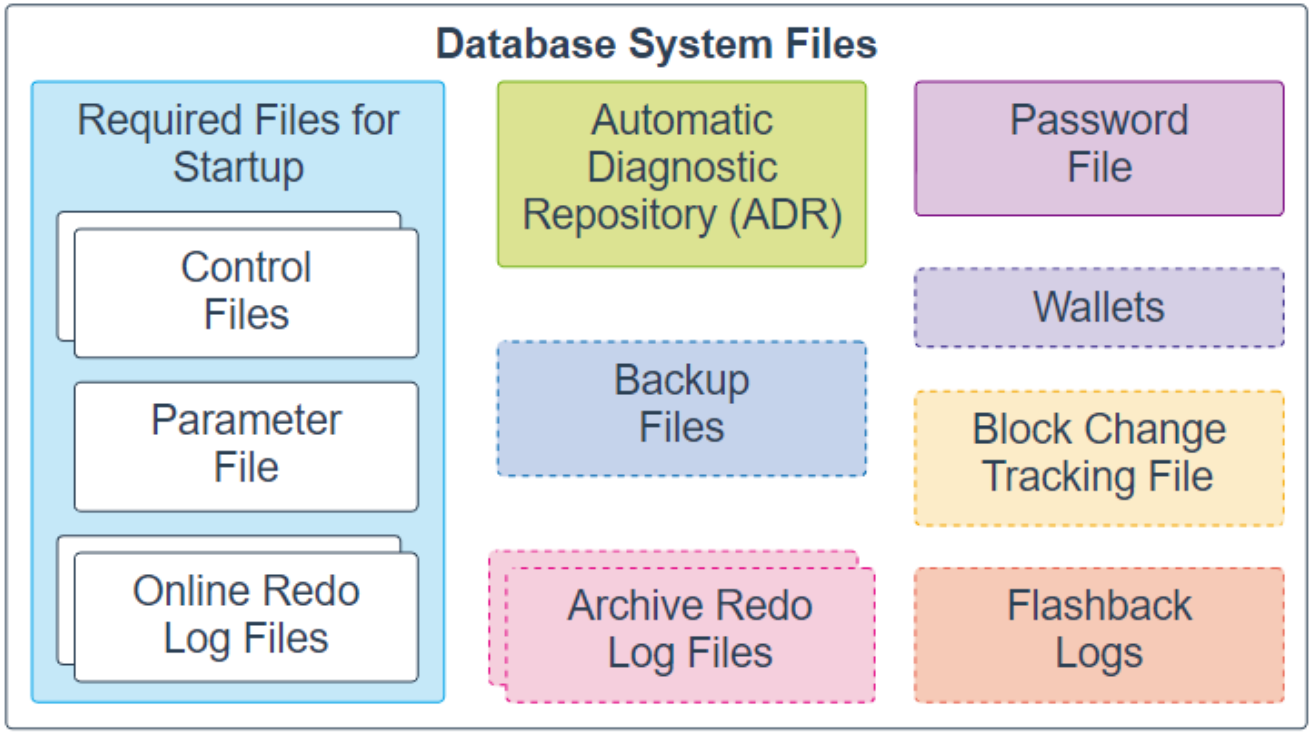
- Instanz
  - Die Instanz besteht aus einem gemeinsam genutzten Speicherbereich, der sogenannten System Global Area (SGA), und einer Reihe von Hintergrundprozessen.
  - ORACLE_SID ("System Identifier")



Quelle: oracle.com

23 DATEIEN (1)

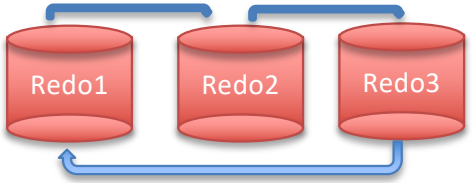
- Datenbank-Dateien
  (logisch aufgeteilt in Tablespaces)
- Online Redo Log-Dateien
- Archivierte Redolog-Dateien
- Control-Dateien
- Parameter-Datei
- ...
- Und mehr



Quelle: oracle.com

24 DATEIEN (2)

- Datenbank-Dateien
  - Enthalten die eigentlichen Daten
  - Sind in Blöcken organisiert (übliche Größe 8 KB)
  - Sind in Tablespaces aufgeteilt ("Platz für Tabellen", logische Aufteilung der Daten)
- Online-Redolog—Dateien ("Transaktionslog")
  - Enthalten die Transaktionsdaten
  - Sind in Gruppen organisiert und werden zyklisch beschrieben
- Archivierte Redolog-Dateien
  - Wichtig für's Recovery



25 DATEIEN (3)

- Control-Dateien
  - Informationen über die physische Struktur der Datenbank (welche Dateien gehören zur DB?)
- Konfigurationsdateien
  - Parameter

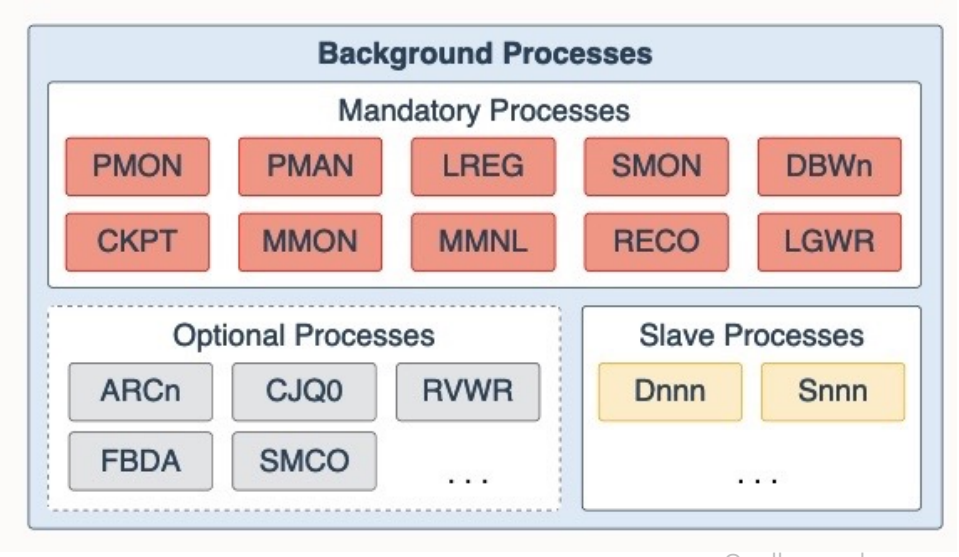
26 SPEZIELLE TABLESPACES

- SYSTEM
  - Enthält die internen Tabellen (sog. Data Dictionary)
- SYSAUX
  - Enthält (z.B.) Tabellen für Performance-Auswertungen
- UNDO
  - Enthält das sog. "Before-Image" bei Updates, wichtig für Lesekonsistenz und "Rollback"
- TEMP
  - Auslagerung von temporären Daten, z.B. bei großen Sortiervorgängen

27 WOFÜR BRAUCHEN WIR "UNDO"?

- Bei einer ändernden Transaktion wird der alte Stand ("Before-Image") des Datensatzes im UNDO-Tablespace abgespeichert.
- Wird die Transaktion abgebrochen (Systemfehler oder "ROLLBACK"), dann wird der alte Stand auf Basis der UNDO-Daten wiederhergestellt.
- Wenn eine andere Transaktion den alten Stand der Daten lesen muss (Stichwort "Lesekonsistenz" / "read-committed"), dann wird dieser alte Stand aus den UNDO-Daten rekonstruiert.
- Andere RDBMS lösen dieses Problem anders, indem sie z.B. bei Änderungen eine komplett neue Version eines Datensatzes anlegen (PostgreSQL)
  - Jeder Datensatz hat Gültigkeitsinformationen (gültig von .. bis)

28 PROZESSE



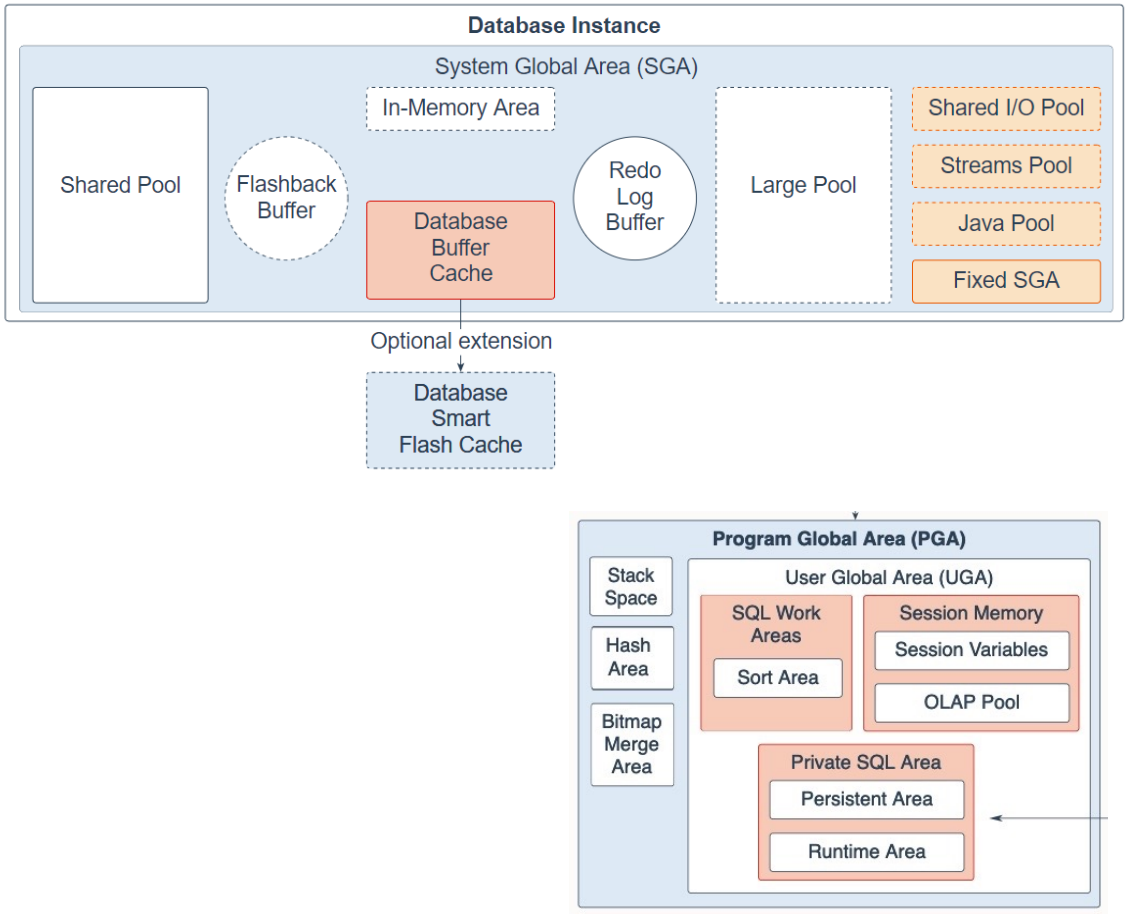
Quelle: oracle.com

- Hintergrundprozesse
  - Eine "Standard-Instanz" hat mehr als 30 Hintergrundprozesse
  - Für das Grundverständnis sind folgende Prozesse wichtig:

SMON	System-Monitor
PMON	Prozess-Monitor – räumt auf, wenn Datenbank-Sitzung abstürzen
LGWR	Log-Writer – schreibt Transaktionsdaten in die Online-Redolog-Dateien
ARC0	Archiver – archiviert volle Online-Redolog-Dateien
DBWR	Database Writer – schreib geänderte Blöcke (asynchron) in die Datenbank-Dateien
CKPT	Checkpointer – löst einen Checkpoint aus (=> alle DB-Dateien konsistent)

29 SPEICHERSTRUKTUREN

- SGA = Shared Global Area (für alle Sitzungen)
  - Buffer Cache (= Cache für DB-Inhalte)
  - Shared Pool (Statement-Cache)
  - Redolog-Buffer (Transaktionsformationen)
  - ...
- PGA = Program Global Area (pro Sitzung)
  - Session-Variablen
  - Speicherplatz zum Sortieren
  - ...



Quelle: oracle.com

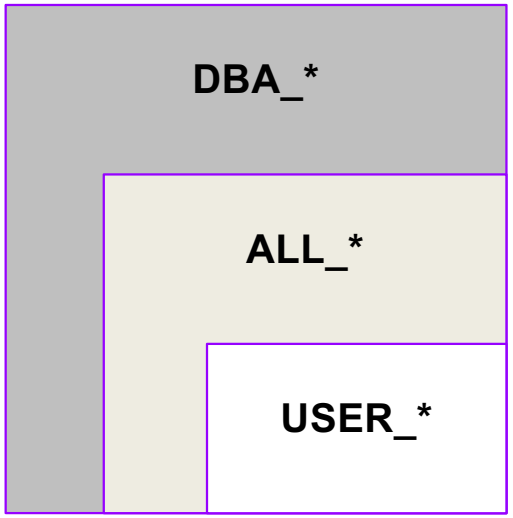
30 DATA DICTIONARY ("SYSTEM CATALOG") (1)

- Jede Datenbank verwaltet sich selbst über interne Tabellen ("Data Dictionary", "System Catalog")

- Das Data Dictionary ist eine Sammlung von Tabellen in der Datenbank,
  - .. die Metadaten zu den Applikationstabellen, Datenbank-Benutzern etc. enthält
  - .. die mit SQL abgefragt werden können wie jede andere Tabelle in der Datenbank
  - .. die vom RDBMS automatisch gepflegt wird

- Beispiele
  - DBA_USERS
  - USER_TABLES
  - USER_VIEWS
  - ALL_TAB_COLUMNS
  - ...

31 DATA DICTIONARY ("SYSTEM CATALOG") (2)



- Im Oracle Data Dictionary gibt es drei Ebenen
  - DBA_* -
  - ALL_* -
  - USER_*
- Beispiel
  - USER_OBJECTS = meine Objekte
  - ALL_OBJECTS = alle Objekte auf die ich Zugriff habe
  - DBA_OBJECTS = der DBA sieht alle Objekte in der Datenbank

WAS PASSIERT BEI EINER TRANSAKTION?

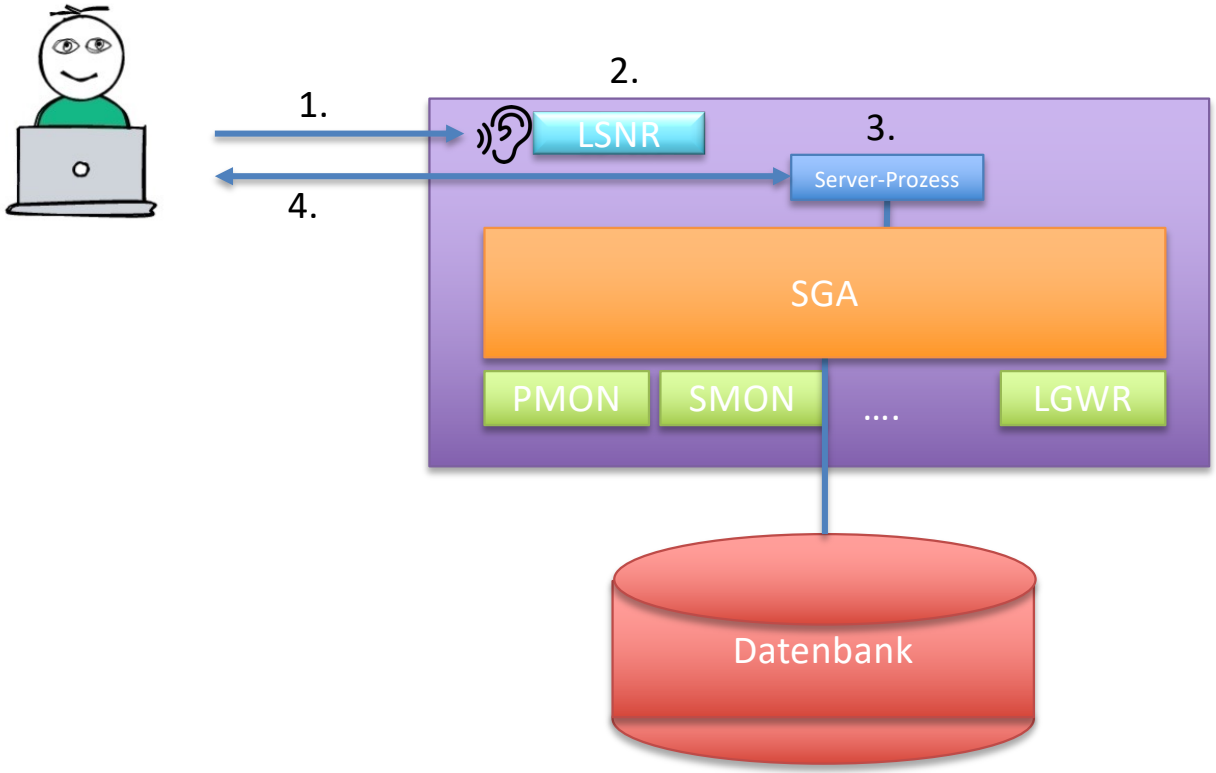
- Ein Benutzer meldet sich an der Datenbank an und führt folgende Befehle aus

```
SQL> UPDATE emp SET ename='KONG' WHERE ename='KING';
SQL> COMMIT;
```

- Was passiert in der Datenbank?

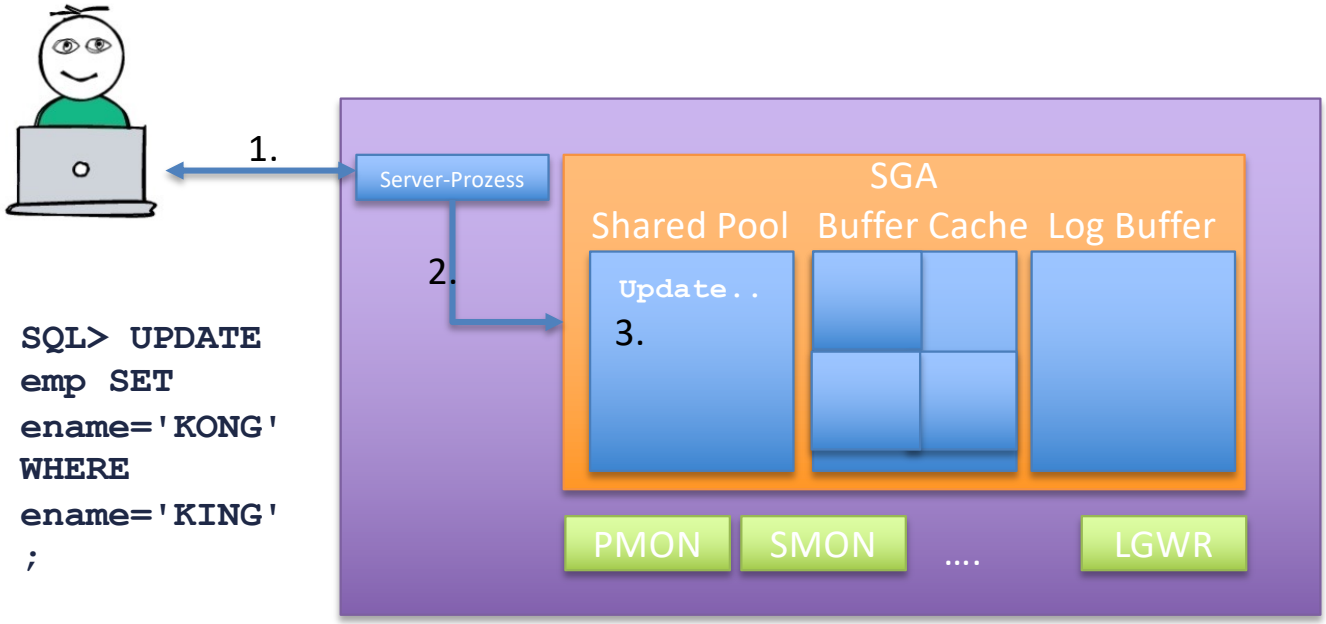
Disclaimer: Alles etwas vereinfacht.

34 SCHRITT 1: DER BENUTZER MELDET SICH AN



1. Der Client schickt eine Verbindungsanfrage an den Server (Host, Port, DB, User, Password)
2. Der Listener nimmt die Anfrage entgegen.
3. Der Listener startet einen Prozess auf dem DB-Server, der sich mit der DB-Instanz "verbindet"
4. Die weitere Kommunikation Client ←→ DB läuft dann über den Server-Prozess.

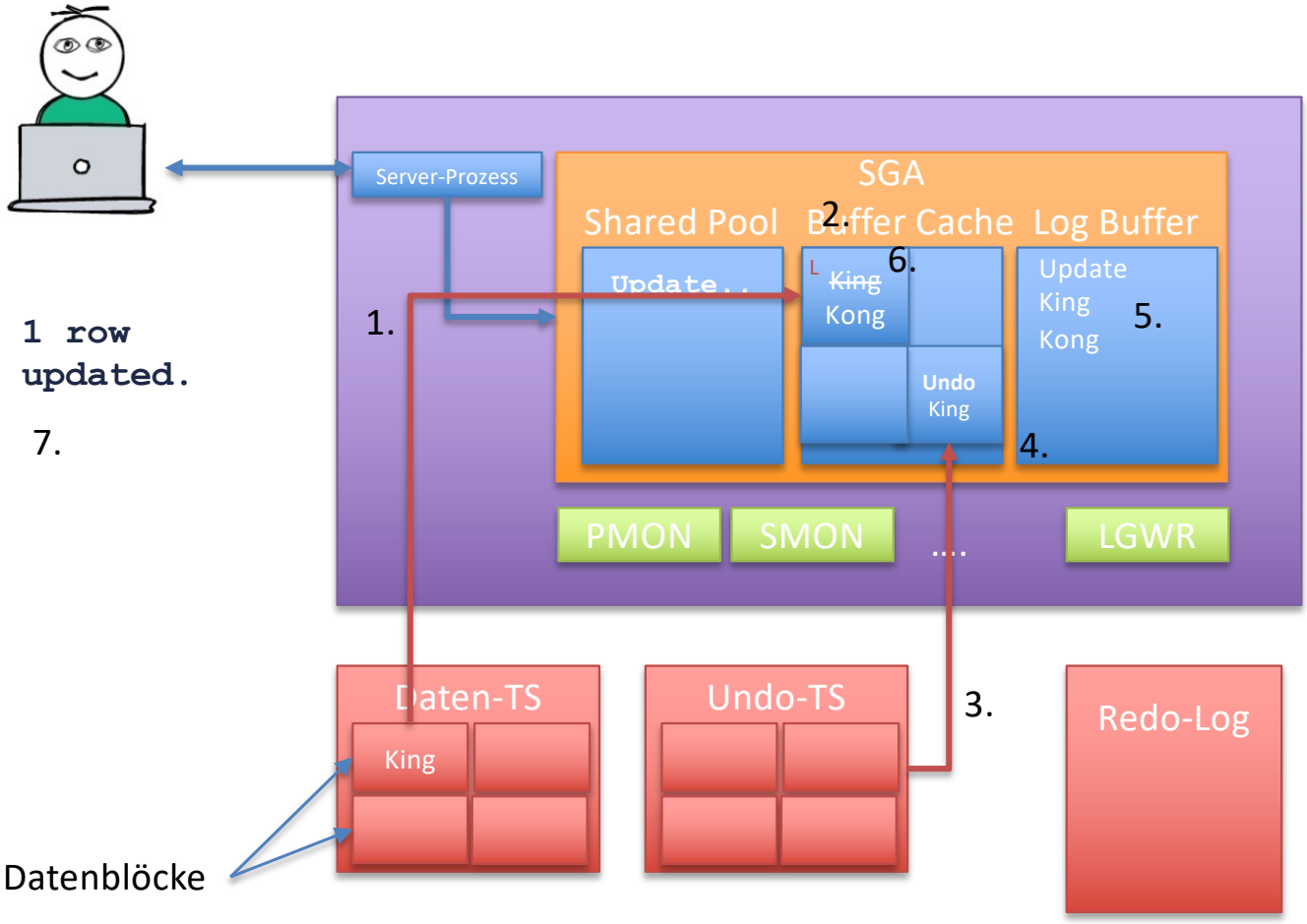
35 SCHRITT 2: DER BENUTZER SETZT DEN UPDATE-BEFEHL AB




```
SQL> UPDATE
emp SET
ename='KONG'
WHERE
ename='KING'
;
```

1. Der Benutzer setzt den Update-Befehl ab
2. Der Befehl wird geparst (Syntax korrekt? Gibt es die Tabelle? Hat der Benutzer die nötigen Rechte?) und ein Execution Plan (wie komme ich an die Daten?) wird erzeugt.
3. Der Befehl + Execution Plan wird im Shared Pool (Statement Cache) abgelegt

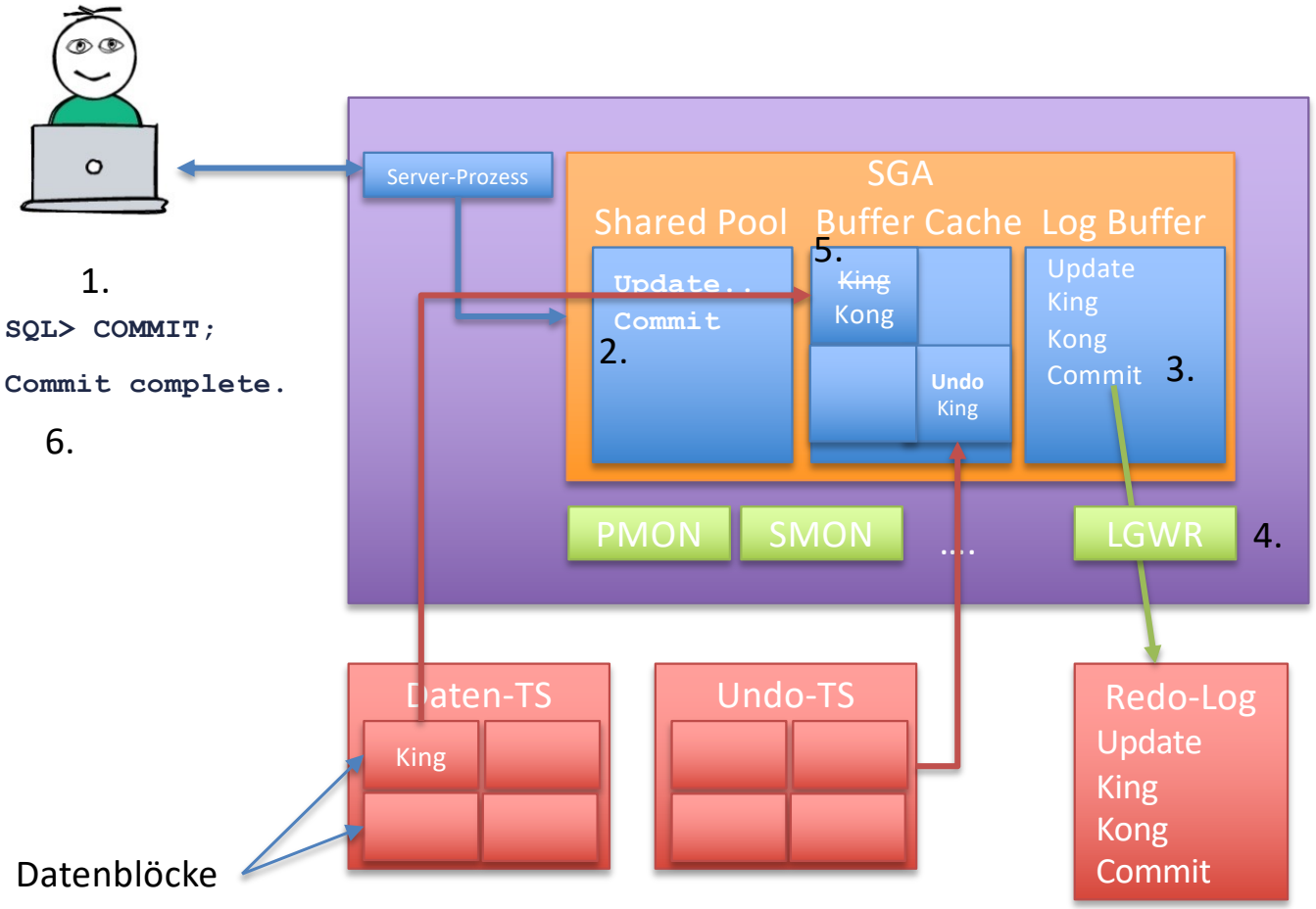
36 SCHRITT 3: DER BEFEHL WIRD AUSGEFÜHRT



1. Der Server-Prozess liest den Block mit dem Datensatz in den Cache
2. Der Server-Prozess sperrt den Datensatz.
3. Der Server-Prozess liest einen Undo-Block in den Cache
4. Der Server-Prozess schreibt den alten Inhalt in den Undo-Block
5. Alter und neuer Inhalt kommen in den Log-Buffer
6. Der Inhalt wird im Cache geändert.
7. Meldung "1 row updated."

trivadis
Part of Accenture

37 SCHRITT 4: DER BENUTZER BESTÄTIGT DIE ÄNDERUNG MIT "COMMIT"



1. Der User bestätigt die Änderung mit "COMMIT"
2. "COMMIT" wird geparst und im Shared-Pool abgelegt
3. Commit—Information wird im Log-Buffer abgelegt
4. Der Log-Writer schreibt die Informationen aus dem Log-Buffer in die Online-Redolog-Datei.
5. Der Lock des Datensatzes wird freigegeben.
6. Der Benutzer bekommt die Meldung "Commit complete."

WAS FEHLT NOCH?

- Der geänderte Datenbank-Block ist bis jetzt nur im Cache geändert, aber (noch) nicht in der Datenbank-Datei!
- Der Database Writer (DBWn) schreibt asynchron
- Wichtig für "Commit Complete" ist: die Änderungen sind in den Redolog-Dateien (Transaktionslog) gespeichert

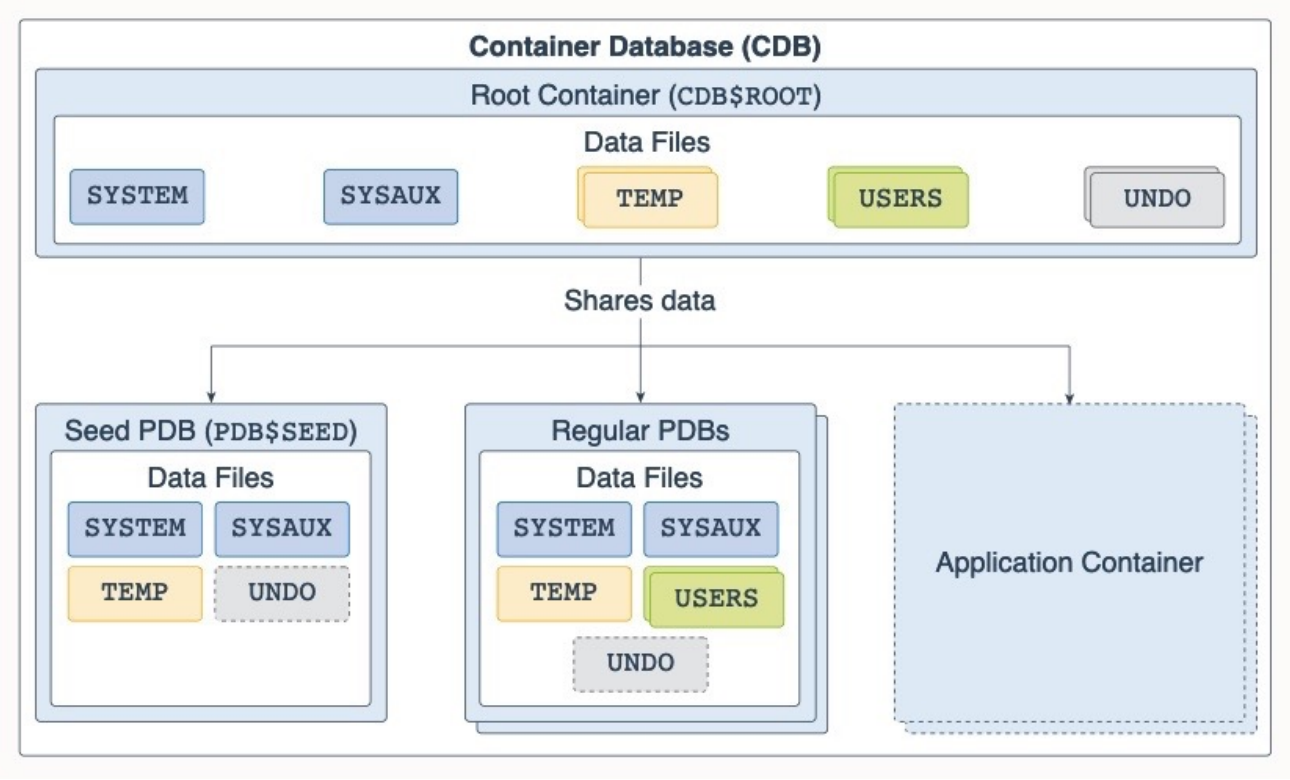
39 UND SONST NOCH ...

- Wenn eine Online-Redolog-Datei voll ist, dann
  - Wechselt die DB in die nächste Redolog-Datei ("Log-Switch")
  - Die volle Redolog-Datei wird archiviert (dafür gibt es den "Archiver-Prozess")
  - Es wird ein sog. "Checkpoint" ausgelöst
    - ➔ alle geänderten Blöcke ("dirty blocks") werden vom Cache in die DB-Dateien geschrieben
    - ➔ die Datenbank-Dateien sind wieder konsistent

ORACLE CONTAINER-DATENBANKEN

41 CONTAINER-DATENBANKEN (1) - ÜBERBLICK

- "Container-Datenbanken" sind Oracle-Datenbanken, die "virtuelle Datenbanken" (sog. "Pluggable Datenbanken" (PDB)) enthalten
- Die PDBs sind in sich abgeschlossen.
- Hintergrund-Prozesse und Speicher-Strukturen (SGA) werden gemeinsam von allen PDBs genutzt
  - ressourcen-sparender
  - "manage many as one"
  - optimal für die Cloud



Quelle: oracle.com

- Kein Unterschiede aus Applikationssicht.
- Die Prinzipien aus dem vorherigen Abschnitt gelten weiterhin.

42 CONTAINER-DATENBANKEN (2) - ARCHITEKTUR

- Spezielle Container
  - CDB$ROOT = enthält die Definitionen des Dictionaries, Metadaten der PDBs
  - PDB$SEED = "Kopiervorlage" für die PDBs

- Data Dictionary
  - Die Strukturen des Dictionaries sind in CDB$ROOT abgelegt
  - Aus den PDBs heraus gibt es Pointer auf die Objekte in CDB$ROOT
  - Die Inhalte des Dictionaries (welche User gibt es in der PDB, welche Tabellen, etc.) werden lokal in der PDB gespeichert

LINKS

44 LINKS

- Ebook (kostenfrei, Registrierung erforderlich)
  "An Introduction to Relational Database Theory"
  https://bookboon.com/en/an-introduction-to-relational-database-theory-ebook

- DB-Engines: DB-Enzyklopädie: https://db-engines.com/de/articles

- Oracle Database 21c Technical Architecture:
  https://docs.oracle.com/en/database/oracle/oracle-database/21/dbiad/index.html

FRAGEN & ANTWORTEN

MARKUS FLECHTNER

- Markus.flechtner@trivadis.com
- Twitter @markusdba
- Blog: markusdba.net|.de

trivadis Part of Accenture

TOGETHER WE ARE #1 PARTNER FOR BUSINESSES TO HARNESS THE POWER OF DATA FOR A SMARTER LIFE

trivadis
Part of Accenture


trivadis
Part of Accenture