

Oracle Fundamentals



Database Concepts

SUMÁRIO

O QUE SÃO DADOS?	3
1.1. DADOS ESTRUTURADOS	3
1.2. DADOS SEMI ESTRUTURADOS	3
1.3. DADOS NÃO ESTRUTURADOS	4
1.4. DADOS VS INFORMAÇÃO	4
1.5. RECURSOS ADICIONAIS	4
2. DATABASES RELACIONAIS	5
2.1. PRINCIPAIS BANCOS RELACIONAIS	5
2.2. RELACIONAMENTO ENTRE TABELAS	5
2.3. VANTAGENS	6
2.4. DESVANTAGENS	6
2.5. ACID	6
2.5.1. ATOMICIDADE	6
2.5.2. CONSISTÊNCIA	7
2.5.3. ISOLAMENTO	7
2.5.4. DURABILIDADE	7
2.6. RECURSOS ADICIONAIS	7
3. PRINCIPAIS OBJETOS DO BANCO DE DADOS	8
3.1. TABELAS	8
3.2. ÍNDICES	8
3.3. VIEWS	9
3.4. PROCEDURES	9
3.5. FUNCTIONS	9
3.6. PACKAGES	10
3.7. TRIGGERS	10
3.8. RECURSOS ADICIONAIS	10
4. ORACLE OVERVIEW	11
4.1. HISTÓRIA	11
4.2. VERSÕES	11
4.3. SUPORTE	13
4.4. VALOR DE MERCADO	13
4.5. RECURSOS ADICIONAIS	13
5. ORACLE VERSIONS	14
5.1. VERSÕES TRADICIONAIS	14
5.2. DATABASE AS A SERVICE	15
5.3. AUTONOMOUS DATABASE	15
5.4. EXADATA (EXACC E EXACS)	16
5.4.1. EXACS	16
5.4.2. EXACC	16
5.5. RECURSOS ADICIONAIS	16
6. TAREFAS COMUNS DE UM DBA	17
6.1. PAPEL DO DBA	17
6.2. RESPONSABILIDADES	17
6.3. CONCLUSÃO	18
6.4. RECURSOS ADICIONAIS	18

1 O QUE SÃO DADOS?

Dados são uma coleção de fatos e informações que podem ser armazenados, processados e analisados para obter insights e tomada de decisão.

Os dados podem ser:

- Dados Estruturados
- Dados semiestruturados
- Dados não estruturados

1.1.DADOS ESTRUTURADOS

Os dados estruturados são organizados em um formato predefinido, como tabelas em um banco de dados relacional. Eles são fáceis de armazenar, acessar e analisar.

ORDER_ID	CUSTOMER_ID	STATUS	SALESMAN_ID	ORDER_DATE	ITEM_ID	PRODUCT_ID	QUANTITY	UNIT_PRICE
88	6	Shipped	61	01-NOV-17	2	11	106	2015.11
88	6	Shipped	61	01-NOV-17	1	278	139	677.99
94	1	Shipped	62	27-OCT-17	6	4	111	2699.99
94	1	Shipped	62	27-OCT-17	5	181	143	999.99
94	1	Shipped	62	27-OCT-17	8	258	73	57.98
94	1	Shipped	62	27-OCT-17	4	80	133	564.89
94	1	Shipped	62	27-OCT-17	7	172	37	358.49
94	1	Shipped	62	27-OCT-17	2	186	146	1449.98
94	1	Shipped	62	27-OCT-17	3	218	86	1388.89
94	1	Shipped	62	27-OCT-17	9	12	33	824.98
94	1	Shipped	62	27-OCT-17	1	255	38	90.99
1	4	Pending	56	15-OCT-17	10	64	147	525.99

1.1 DADOS SEMI ESTRUTURADOS

Já os dados semiestruturados, como XML ou JSON, têm alguma estrutura, mas não seguem um formato rígido. Eles são usados principalmente para troca de dados entre aplicativos ou sistemas.

```

1 {
2   "count": 7,
3   "items": ["socks", "pants", "shirts", "hats"],
4   "manufacturer": {
5     "name": "Molly's Seamstress Shop",
6     "id": 39233,
7     "location": {
8       "address": "123 Pickleton Dr.",
9       "city": "Tucson",
10      "state": "AZ",
11      "zip": 85705
12    }
13  },
14  "total_price": "$393.23",
15  "purchase_date": "2022-05-30",
16  "country": "USA"
17 }
```

1.2 DADOS NÃO ESTRUTURADOS

Os dados não estruturados, como imagens, vídeos e textos, não têm uma estrutura definida e são mais difíceis de processar e analisar.



1.3 DADOS VS INFORMAÇÃO

Dado e informação são termos relacionados, mas têm significados diferentes. Dados são fatos brutos, como números, nomes, datas ou outras informações que podem ser coletadas e armazenadas. Por exemplo, um conjunto de dados pode incluir as vendas diárias de um produto em um determinado período de tempo.

Já a informação é a interpretação ou significado atribuído aos dados. A informação é derivada dos dados por meio de análise, interpretação ou processamento. Por exemplo, se você analisar os dados de vendas diárias e calcular a média de vendas por dia, isso se torna uma informação útil que pode ajudar na tomada de decisões de negócios.

Em resumo, os dados são o que coletamos e armazenamos, enquanto a informação é o que fazemos com esses dados. A informação é criada a partir de dados por meio de processamento, análise e interpretação.



1.4 RECURSOS ADICIONAIS

- Leia o capítulo e escute o áudio ao mesmo tempo
- Importe o deck “01 - What is Data”
- Revise os flashcards

2 DATABASES RELACIONAIS

Um banco de dados relacional é um tipo de armazenamento de informações que usa tabelas para guardar dados relacionados.

Cada tabela representa algo, como por exemplo uma lista de clientes ou uma lista de produtos, e as informações nessas tabelas podem ser conectadas entre si para formar relações. Isso ajuda a manter os dados organizados e a torná-los fáceis de encontrar e atualizar.

2.1 PRINCIPAIS BANCOS RELACIONAIS

Os principais bancos relacionais são:

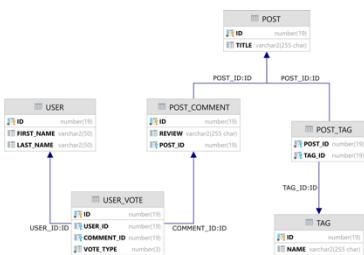
1. Oracle
2. MySQL
3. Sql Server
4. Postgres






2.2 RELACIONAMENTO ENTRE TABELAS

As tabelas são relacionadas umas com as outras através de chaves primárias e chaves estrangeiras. Uma chave primária é uma coluna ou um conjunto de colunas que identificam exclusivamente cada linha em uma tabela. Uma chave estrangeira é uma coluna em uma tabela que se refere à chave primária de outra tabela.



2.3 VANTAGENS

Os bancos de dados relacionais oferecem muitas vantagens. Uma das principais é a capacidade de realizar **consultas complexas** e eficientes.

Por exemplo, você pode recuperar informações de várias tabelas relacionadas com uma única consulta SQL. Isso é conhecido como JOIN e é uma das principais vantagens dos bancos de dados relacionais.

Os bancos de dados relacionais também oferecem a capacidade de manter a **integridade dos dados**. As restrições de integridade garantem que os dados em uma tabela estejam dentro de limites aceitáveis.

Por exemplo, você pode definir uma restrição de integridade para garantir que um valor na tabela de pedidos não possa ser inserido a menos que corresponda a um valor na tabela de clientes.

2.4 DESVANTAGENS

No entanto, os bancos de dados relacionais também têm algumas desvantagens. Um dos principais problemas é a **escalabilidade**.

À medida que a quantidade de dados aumenta, a **complexidade** das consultas também aumenta, o que pode levar a **problemas de desempenho**. Além disso, a estrutura rígida das tabelas relacionais pode tornar difícil lidar com tipos de dados complexos ou com dados sem uma estrutura clara.

2.5 ACID

ACID é um conjunto de regras que garantem que as transações em um banco de dados sejam executadas de forma confiável.

ACID é um acrônimo:

- Atomicidade
- Consistência
- Isolamento
- Durabilidade

1.1.1. ATOMICIDADE

Atomicidade significa que uma transação é vista como uma **operação única**, ou seja, ou todas as partes da transação são executadas ou nenhuma é.

 Por exemplo, imagine que você queira transferir dinheiro de uma conta para outra. Para garantir a atomicidade da transação, o sistema deve executar duas operações como uma única transação: retirar o dinheiro da conta de origem e depositar o mesmo valor na conta de destino. Se uma das operações falhar, a transação deve ser revertida e as contas voltam ao estado anterior à transação.

Isso garante que a transação seja executada corretamente e que não haja perda de dinheiro ou dados em caso de falha durante a transferência. A

atomicidade é uma das propriedades essenciais das transações em bancos de dados.

1.1.2. CONSISTÊNCIA

Consistência significa que os dados devem estar **consistentes antes e depois da transação**, ou seja, a transação não pode violar as regras e restrições que garantem a integridade dos dados.

💡 Por exemplo, imagine que você tem uma tabela de clientes em que a idade é um campo obrigatório e deve ser um número inteiro positivo. Se uma transação tentar inserir um cliente com idade negativa ou deixar o campo em branco, a transação deve ser revertida, pois isso violaria a consistência dos dados.

Isso garante que os dados do banco de dados estejam sempre **confiáveis** e úteis para quem os usa, já que os dados são consistentes e obedecem às regras do sistema. A consistência é uma das propriedades essenciais das transações em bancos de dados.

1.1.3. ISOLAMENTO

Isolamento significa que uma transação deve ser executada sem **interferência** de outras transações em andamento, ou seja, cada transação deve parecer que **está sendo executada sozinha** no banco de dados.

💡 Por exemplo, imagine que duas transações estejam sendo executadas ao mesmo tempo, ambas tentando acessar o mesmo registro de uma tabela. O isolamento garante que cada transação veja apenas o valor original do registro, como se estivesse executando sozinha no banco de dados, mesmo que outras transações estejam acessando os mesmos dados.

Isso garante que as transações não interfiram umas nas outras, impedindo conflitos de dados ou informações inconsistentes. O isolamento é uma das propriedades essenciais das transações em bancos de dados.

1.1.4. DURABILIDADE

Durabilidade significa que uma vez que uma transação é concluída com sucesso, os dados são permanentemente armazenados e não serão perdidos, mesmo em caso de falhas no sistema.

💡 Por exemplo, imagine que você tenha concluído uma transação de atualização de dados em um banco de dados, mas em seguida ocorreu uma queda de energia. Mesmo após a falha no sistema, a atualização de dados deve ser mantida e permanecer salva no banco de dados, mesmo que o sistema precise ser reiniciado.

Isso garante que os dados permaneçam seguros e permanentemente armazenados, mesmo em caso de falhas no sistema. A durabilidade é uma das propriedades essenciais das transações em bancos de dados.

2.6 RECURSOS ADICIONAIS

- Leia o capítulo e escute o áudio ao mesmo tempo
- Importe o deck “02 - Database Concepts”
- Revise os flashcards

3 PRINCIPAIS OBJETOS DO BANCO DE DADOS

Um banco de dados é uma coleção organizada de dados que são armazenados e gerenciados por um sistema de gerenciamento de banco de dados (SGBD). Esses dados são organizados em objetos de banco de dados, que são as unidades básicas de armazenamento no banco de dados.

- Tabelas
- Índices
- Views
- Procedures
- Functions
- Packages
- Triggers

3.1 TABELAS

As tabelas são os objetos de banco de dados mais básicos e contêm os dados do banco de dados em si. Elas são compostas por colunas e linhas, onde cada coluna representa um tipo de dado específico e cada linha representa uma entrada de dados.

```
create table dept(
    deptno number(2,0),
    dname  varchar2(15),
    loc    varchar(15),
    constraint pk_dept primary key (deptno)
);

create table emp(
    empno  number(4,0),
    ename  varchar2(10),
    job    varchar(9),
    mgr    number(4,0),
    hiredate date,
    sal    number(7,2),
    comm   number(7,2),
    deptno number(2,0),
    constraint pk_emp primary key (empno),
    constraint fk_deptno foreign key (deptno) references dept (deptno)
);

insert into dept values(10, 'ACCOUNTING', 'NEW YORK');
insert into dept values(20, 'RESEARCH', 'DALLAS');
insert into dept values(30, 'SALES', 'CHICAGO');
insert into dept values(40, 'OPERATIONS', 'BOSTON');

INSERT INTO emp VALUES (7839, 'KING', 'PRESIDENT', NULL, TO_DATE('17-11-1981'), '00-mm-yyyy'), 5000, NULL, 10;
INSERT INTO emp VALUES (7844, 'MILLER', 'CLERK',    TO_DATE('23-5-1981'), '00-mm-yyyy'), 2950, NULL, 20;
INSERT INTO emp VALUES (7876, 'BLAKE', 'MANAGER', TO_DATE('12-5-1981'), '00-mm-yyyy'), 2450, NULL, 30;
INSERT INTO emp VALUES (7902, 'JONES', 'MANAGER', TO_DATE('2-4-1981'), '00-mm-yyyy'), 2975, NULL, 20;
INSERT INTO emp VALUES (7934, 'MARTIN', 'SALESMAN', TO_DATE('12-8-1981'), '00-mm-yyyy'), 1250, 500, 30;
INSERT INTO emp VALUES (7962, 'FORD', 'ANALYST',  TO_DATE('3-12-1981'), '00-mm-yyyy'), 3000, NULL, 20;
INSERT INTO emp VALUES (7965, 'SMITH', 'CLERK',    TO_DATE('17-12-1980'), '00-mm-yyyy'), 800, NULL, 20;
INSERT INTO emp VALUES (7978, 'ADAMS', 'SALESMAN', TO_DATE('2-2-1981'), '00-mm-yyyy'), 1250, 500, 30;
INSERT INTO emp VALUES (7985, 'WARD', 'SALESMAN', TO_DATE('22-2-1981'), '00-mm-yyyy'), 1250, 500, 30;
INSERT INTO emp VALUES (7986, 'TOMAS', 'SALESMAN', TO_DATE('28-9-1981'), '00-mm-yyyy'), 1250, 1400, 30;
INSERT INTO emp VALUES (7996, 'JAMES', 'CLERK',    TO_DATE('3-12-1981'), '00-mm-yyyy'), 950, NULL, 20;
INSERT INTO emp VALUES (7975, 'ADAMS', 'CLERK',    TO_DATE('13-3-1981'), '00-mm-rrr'), -51, 1180, NULL, 20;
INSERT INTO emp VALUES (7994, 'MILLER', 'CLERK',    TO_DATE('23-1-1982'), '00-mm-yyyy'), 1300, NULL, 10;

COMMIT;
```

3.2 ÍNDICES

Os índices são estruturas que ajudam a acelerar a recuperação de dados em tabelas grandes. Eles são criados em colunas específicas de tabelas e armazenam informações sobre a localização dos dados dentro da tabela.

```
CREATE TABLE EMPLOYEES (
    EMP_ID NUMBER(6) NOT NULL,
    FIRST_NAME VARCHAR2(20),
    LAST_NAME VARCHAR2(25),
    EMAIL    VARCHAR2(25),
    PHONE_NUMBER VARCHAR2(20),
    HIRE_DATE DATE,
    JOB_ID  VARCHAR2(10),
    SALARY   NUMBER(8,2),
    MANAGER_ID NUMBER(6),
    DEPARTMENT_ID NUMBER(4)
);

CREATE INDEX idx_lastname
ON EMPLOYEES (LAST_NAME);
```

3.3 VIEWS

Views são consultas salvas que podem ser acessadas como se fossem tabelas reais. Elas são criadas a partir de consultas SQL complexas e podem ser usadas para simplificar o acesso a dados em uma ou mais tabelas.

```
create or replace view VW_EMP_DEPT
AS
SELECT
    b.ename,
    b.job,
    b.sal,
    a.dname,
    a.loc
FROM
    dept a
INNER JOIN emp b ON a.deptno = b.deptno
WHERE
    job = 'MANAGER'
ORDER BY
    sal DESC;

SELECT * FROM VW_EMP_DEPT;
```

3.4 PROCEDURES

Stored Procedures são rotinas de banco de dados que são escritas em uma linguagem específica do SGBD. Eles podem ser usados para executar operações complexas no banco de dados e automatizar tarefas.

```
create or replace PROCEDURE Get_emp_names (Dept_num IN NUMBER) IS
    Emp_name      VARCHAR2(10);
    CURSOR        c1 (Depno NUMBER) IS
        SELECT Ename FROM emp
        WHERE deptno = Depno;
BEGIN
    OPEN c1(Dept_num);
    LOOP
        FETCH c1 INTO Emp_name;
        EXIT WHEN C1%NOTFOUND;
        DBMS_OUTPUT.PUT_LINE(Emp_name);
    END LOOP;
    CLOSE c1;
END;

select * from emp;
exec Get_emp_names (10);
```

3.5 FUNCTIONS

Function é um bloco de código em programação que recebe um ou mais valores de entrada, executa uma série de instruções e retorna um valor específico como resultado. Ela pode ser chamada em diferentes partes do programa para executar uma tarefa específica. As funções ajudam a modularizar o código, tornando-o mais organizado, reutilizável e fácil de manter.

```
CREATE OR REPLACE FUNCTION calcularSalarioAnual(salarioMensal NUMBER) RETURN NUMBER IS
    salarioAnual NUMBER;
BEGIN
    salarioAnual := salarioMensal * 12; -- Multiplica o salário mensal por 12 para obter o salário anual
    RETURN salarioAnual; -- Retorna o salário anual calculado
END;
/

select calcularSalarioAnual(200) from dual;
```

3.6 PACKAGES

Package é uma estrutura que permite agrupar objetos relacionados, como procedimentos, funções, variáveis e tipos de dados, em um único objeto. Ele fornece uma forma organizada de armazenar e gerenciar a lógica do programa e os dados associados a ele.

```
-- Especificação do Package
CREATE OR REPLACE PACKAGE simple_package AS
  -- Declaração da função
  FUNCTION add_numbers(num1 NUMBER, num2 NUMBER) RETURN NUMBER;
  -- Declaração da procedure
  PROCEDURE print_hello_world;
END simple_package;
/

-- Corpo do Package
CREATE OR REPLACE PACKAGE BODY simple_package AS
  -- Implementação da função
  FUNCTION add_numbers(num1 NUMBER, num2 NUMBER) RETURN NUMBER AS
  BEGIN
    RETURN num1 + num2;
  END add_numbers;

  -- Implementação da procedure
  PROCEDURE print_hello_world AS
  BEGIN
    DBMS_OUTPUT.PUT_LINE('Hello, world!');
  END print_hello_world;
END simple_package;
/

-- Executando a function do package
select simple_package.add_numbers(1,2) from dual;
-- Executando a procedure do package
set serveroutput on
exec simple_package.print_hello_world;
```

3.7 TRIGGERS

As triggers são rotinas de banco de dados que são executadas automaticamente quando ocorre um evento específico no banco de dados, como uma atualização ou exclusão de dados. Eles são usados para aplicar regras de negócios e garantir a consistência dos dados.

```
DROP TABLE Emp_log;
CREATE TABLE Emp_log (
  Emp_id      NUMBER,
  Log_date    DATE,
  New_salary  NUMBER,
  Action       VARCHAR2(20),
  host varchar2(100),
  ip varchar2(100)
);

DROP TRIGGER log_salary_increase;
CREATE OR REPLACE TRIGGER log_salary_increase
  AFTER UPDATE OF sal ON emp
  FOR EACH ROW
BEGIN
  INSERT INTO Emp_log (Emp_id, Log_date, New_salary, Action,host,ip)
  VALUES (:NEW.empno, SYSDATE, :NEW.sal, 'New Salary',sys_context ('USERENV','HOST'),sys_context ('USERENV','IP_ADDRESS'));
END;
/

select * from emp_log;
select * from emp;

update emp
set sal = 10000
where ename = 'SCOTT';
```

3.8 RECURSOS ADICIONAIS

- Leia o capítulo e escute o áudio ao mesmo tempo
- Importe o deck “03 - Introduction to Database Objects”
- Revise os flashcards

4 ORACLE OVERVIEW

4.1 HISTÓRIA

A Oracle foi fundada em 16 de junho de 1977 por Larry Ellison, Bob Miner e Ed Oates. Na época, Ellison era inspirado por um artigo escrito por Edgar F. Codd sobre bancos de dados relacionais, que ele acreditava ser o futuro do banco de dados.

No início, a empresa era conhecida como Software Development Laboratories (SDL). Em 1979, a empresa foi renomeada para Relational Software Inc. (RSI) e finalmente se tornou Oracle Systems Corporation em 1982, em homenagem ao seu produto principal, o Oracle Database.

O Oracle Database, também conhecido como Oracle RDBMS ou simplesmente Oracle, é um sistema de gerenciamento de banco de dados relacional. O Oracle Database foi o primeiro banco de dados comercial a suportar a linguagem SQL, o que contribuiu muito para a sua popularidade.

Com o passar do tempo, a Oracle expandiu seus produtos e serviços para além dos bancos de dados para incluir, entre outros, servidores de middleware, aplicativos de negócios e até mesmo hardware (através da aquisição da Sun Microsystems em 2010). A Oracle também desempenhou um papel importante no desenvolvimento e padronização de tecnologias como Java e Linux.

Hoje, a Oracle é uma gigante global de tecnologia com dezenas de milhares de funcionários em todo o mundo, fornecendo uma ampla variedade de produtos e serviços para empresas de todos os tamanhos e setores.

4.2 VERSÕES

ORACLE V2 (1979): Foi a primeira versão comercial. Disponível com a linguagem SQL foi um marco na história dos banco de dados relacionais.

ORACLE V3 (1983): Primeira versão a rodar em mainframes, mini computadores e computadores pessoais. Escrito em linguagem C, permitindo que o banco rodasse em múltiplas plataformas.

ORACLE V4 (1984): Introduziu o recurso de leitura consistente, Lançado em 1984.

ORACLE V5 (1985): Introduziu suporte a cliente servidor e bancos de dados distribuídos.

ORACLE V6 (1986): Fez melhorias em I/O, bloqueio de linha, escalabilidade e recursos de backup e recuperação. Também foi introduzido a primeira versão da linguagem PL/SQL

ORACLE V7 (1992): Introduziu Stored Procedures e Triggers

ORACLE V8 (1997): Passou a suportar vários novos tipos de dados e também passou a suportar particionamento de tabela

ORACLE 8i (1999): Passou a suportar protocolos de internet e suporte para Java, foi desenhado para a internet.

ORACLE 9i (2001): Introduziu o Oracle Real Application Clusters (RAC).

ORACLE 10g (2003): G de Grid Computing. Lançado para habilitar organizações a virtualizar recursos computacionais. Lançado o Auomatic Storage Management (ASM) e o Automatic Workload Repository (AWR)

ORACLE 11g (2007): Introdução de recursos como tabelas em somente leitura, índices invísiveis, colunas virtuais, Active Data Guard, Exadata e melhorias gerais nos recursos existentes.

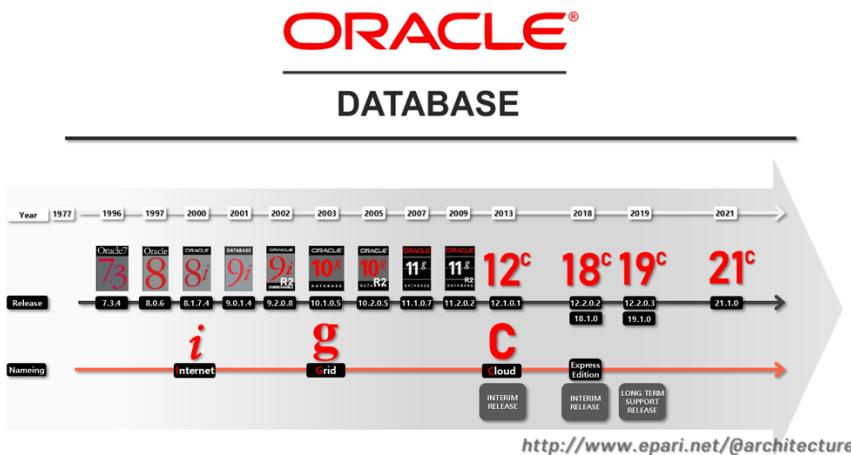
ORACLE 12c (2012): C de Cloud. Preparado para Cloud Computing. A principal característica é a introdução da arquitetura Multitenant.

ORACLE 18c (2018): É uma evolução do Oracle 12c (12.2.0.2)

ORACLE 19c (2019): É uma evolução do 12c (12.2.0.3) e a versão de suporte de longo prazo do Oracle Database 12c. Seu maior foco é na estabilidade e a introdução de muitos pequenos novos recursos.

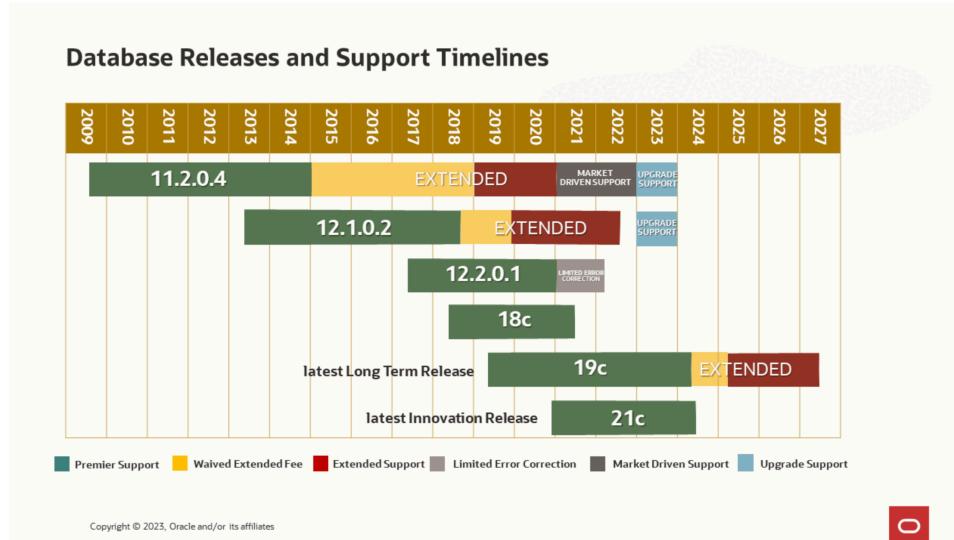
ORACLE 21c (2020): É uma versão de inovação, mas também é uma versão produtiva. Suporta Blockchain Tables e JSON Data Types.

ORACLE 23c (2023): Até o momento só disponibilizou a versão Free Edition.

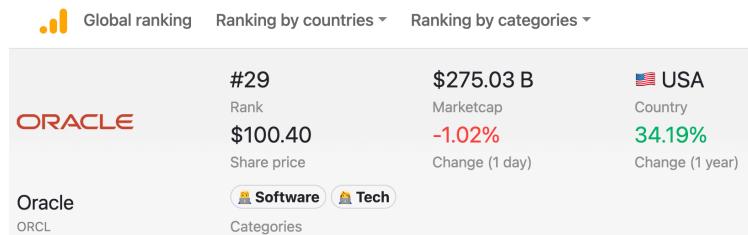


4.3 SUPORTE

Até o momento, as versões com suporte ativo neste momento são: 19c e 21c.



4.4 VALOR DE MERCADO



Stock price history of Oracle from 1986 to 2023



<https://companiesmarketcap.com/oracle/stock-price-history/>

4.5 RECURSOS ADICIONAIS

- Leia o capítulo e escute o áudio ao mesmo tempo

5 ORACLE VERSIONS

O Oracle possui algumas versões diferentes, algumas dessas versões estão disponíveis apenas na OCI. Podemos dizer que o Oracle tem as versões “Tradicionais” que rodam em ambiente On Premisses e as versões que rodam na OCI, que são as versões “Database As A Service” (DBaaS).

No licenciamento do Oracle, além de comprar a versão em si, você pode comprar Extra Options, que são “adicionais” e são licenciados à parte.

Por exemplo, a empresa pode comprar um Oracle Enterprise Edition e uma Extra Option de Partitioning.

As versões do Oracle na OCI, basicamente são variações das versões “Tradicionais” que essas extra options já estão inclusas na versão escolhida.

5.1 VERSÕES TRADICIONAIS

Oracle Database Standard Edition 2: A Standard Edition 2 é uma opção de licenciamento básica e econômica que oferece um conjunto de recursos essenciais do Oracle Database. Ela é adequada para pequenas e médias empresas ou aplicativos que não exigem recursos avançados, como clustering ou recursos de gerenciamento de grandes volumes de dados. A licença SE2 é limitada a um **máximo de 2 soquetes de CPU no servidor**.

Oracle Database Enterprise Edition: A Enterprise Edition é a versão mais abrangente do Oracle Database, fornecendo todos os recursos disponíveis na Standard Edition 2, além de opções adicionais avançadas e de alto desempenho, como particionamento, replicação, alta disponibilidade e escalabilidade. A licença Enterprise Edition é adequada para empresas que exigem recursos mais sofisticados e que possuem aplicativos de **missão crítica**. A quantidade de licenciamento é baseada no número de processadores ou núcleos do servidor.

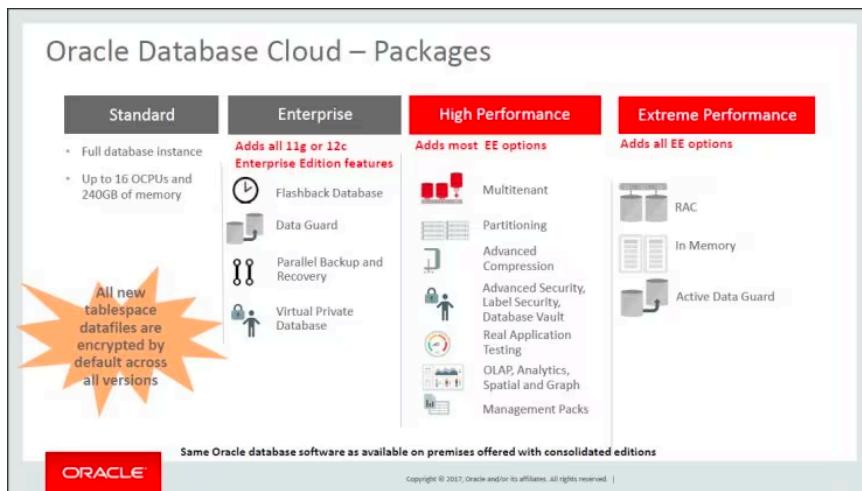
Oracle XE / Free Edition: A Express Edition & Free Edition é uma versão gratuita do Oracle Database, destinada principalmente para desenvolvimento, testes e ambientes de baixa carga. Ela tem algumas limitações de recursos, como restrição de uso de CPU e tamanho do banco de dados, sendo indicada para projetos menores ou para fins de aprendizado.

- Banco de dados de no máximo 12GB (dados do usuário)
- 2 GB de RAM
- Dois processadores
- 1 instalação por servidor

5.2 DATABASE AS A SERVICE

A OCI oferece suporte a vários tipos de bancos de dados, incluindo Oracle Database, MySQL, PostgreSQL e NoSQL. DBAAS na OCI oferece uma maneira fácil e conveniente de implantar e gerenciar bancos de dados na nuvem, permitindo que os clientes se concentrem em seus aplicativos e dados, em vez de se preocuparem com a infraestrutura. Os bancos de dados Oracle como serviço são:

- Oracle Standard (DBCS SE)
- Oracle Enterprise (DBCS EE)
- Oracle High Performance (DBCS EE-HP)
- Oracle Extreme Performance (DBCS EE-EP)



5.3 AUTONOMOUS DATABASE

Oracle Autonomous Database é uma oferta de banco de dados em nuvem da Oracle Corporation, projetada para automatizar muitas das tarefas rotineiras necessárias para gerenciar um banco de dados.

É uma solução de banco de dados em nuvem que busca simplificar o gerenciamento de bancos de dados, oferecendo autonomia, segurança, desempenho e eficiência para as empresas.

- Autonomia
- Elasticidade
- Segurança
- Desempenho
- Redução de custos
- Simplicidade
- Integração com a nuvem

5.4 EXADATA (EXACC E EXACS)

O ExaCC e o ExaCS são dois produtos da Oracle que fazem parte da linha de soluções de banco de dados Oracle Exadata. Embora ambos sejam baseados na tecnologia Exadata, existem diferenças significativas entre eles:

- Exadata Cloud at Customer (**ExaCC**)
- Exadata Cloud Service (**ExaCS**)

1.1.5. EXACS

O ExaCS é uma oferta de banco de dados como serviço (DBaaS) baseada na nuvem pública da Oracle. Ele fornece acesso a uma infraestrutura compartilhada do Exadata, hospedada e gerenciada pela Oracle em seus data centers.

Os clientes podem provisionar instâncias de banco de dados no ExaCS de acordo com suas necessidades e pagar com base no consumo, sem a necessidade de gerenciar a infraestrutura subjacente.

1.1.6. EXACC

O ExaCC é uma solução de infraestrutura de nuvem privada oferecida pela Oracle. Ele é projetado para ser implantado nas instalações do cliente, fornecendo todos os benefícios e recursos do Exadata como serviço.

O ExaCC oferece a flexibilidade e o controle de uma nuvem privada, permitindo que as organizações executem cargas de trabalho críticas no local, enquanto se beneficiam do hardware e da tecnologia do Exadata.

5.5 RECURSOS ADICIONAIS

- Leia o capítulo e escute o áudio ao mesmo tempo
- Importe o deck “05 - Oracle Versions”
- Revise os flashcards

6 TAREFAS COMUNS DE UM DBA

6.1 PAPEL DO DBA

O DBA é a interface entre o banco de dados e seus usuários. Sua principal responsabilidade é garantir a disponibilidade, acessibilidade e segurança do banco de dados.

6.2 RESPONSABILIDADES

Instalação/Upgrade/Patch e Configuração do Banco de Dados: O DBA é responsável pela instalação do banco de dados Oracle e pela configuração adequada do sistema para suportar o banco de dados.

Instalação/Upgrade/Patch e Configuração do Banco de Dados: O DBA é responsável pela instalação do banco de dados Oracle e pela configuração adequada do sistema para suportar o banco de dados.

Administração e Manutenção do Banco de Dados: Uma vez instalado, o DBA gerencia todos os aspectos do banco de dados. Isso pode incluir atualizações, patches, manutenção de índices e garantia de desempenho do banco de dados.

Segurança do banco de dados: O DBA Oracle é responsável pela segurança do banco de dados. Isso inclui a definição de privilégios de usuários, a criação de políticas de segurança e a proteção contra ameaças externas.

Backup & Recovery: O DBA Oracle é responsável por garantir a recuperação de dados em caso de perda de dados. Isso envolve a implementação de estratégias de backup e recuperação e a execução regular de backups.

Performance: Otimização de desempenho: O DBA Oracle trabalha para garantir o desempenho ideal do banco de dados. Isso pode envolver a otimização de consultas, a criação e manutenção de índices e a monitoração do desempenho do banco de dados.

Design do Banco de dados: Embora nem sempre seja responsabilidade do DBA, em algumas organizações, o DBA Oracle também pode estar envolvido no design do banco de dados.

Monitoramento Pro ativo: Utilizando ferramentas de monitoramento, os DBAs identificam e resolvem problemas antes que impactem as operações de negócios, mantendo assim a saúde do banco de dados.

Supor te e resolução de problemas: Como especialistas, os DBAs são o primeiro ponto de contato quando há problemas com o banco de dados. Eles precisam identificar rapidamente a causa raiz e implementar soluções, minimizando o tempo de inatividade e o impacto nos usuários finais.

Plano de contingência: DBAs são responsáveis por planejar o inesperado. Isso envolve a criação e implementação de planos de contingência e recuperação de desastres, para garantir a continuidade dos negócios em caso de falhas ou desastres.

Migração de Dados: DBAs são frequentemente encarregados de migrar dados de um sistema para outro. Eles precisam garantir que os dados sejam transferidos com precisão e eficiência, enquanto minimizam o tempo de inatividade do sistema.

Automação de scripts: DBAs Oracle utilizam scripts e automação para aumentar a eficiência e reduzir a margem de erro em tarefas repetitivas. Isso pode incluir a automação de backups, monitoramento de desempenho, e outras tarefas de manutenção.

Colaboração com outras equipes: O trabalho de um DBA não é isolado. Eles colaboram com desenvolvedores, analistas de dados, gerentes de projeto e outros membros da equipe de TI para alcançar os objetivos da organização. Sua capacidade de se comunicar de forma eficaz e trabalhar como parte de uma equipe é crucial.

Plano de Capacidade: O DBA monitora o crescimento do banco de dados e planeja a capacidade futura, identificando quando será necessário adicionar mais recursos, como espaço de armazenamento, processamento ou memória.

News Features: O DBA deve estar sempre atualizado sobre os novos recursos para apoiar à equipe de sistemas e implementar melhorias no banco de dados.

6.3 CONCLUSÃO

O papel de um DBA Oracle é abrangente, dinâmico e vital para qualquer organização que depende de dados. Eles são solucionadores de problemas, comunicadores, estratégistas e especialistas técnicos, todos em um.

6.4 RECURSOS ADICIONAIS

- Leia o capítulo e escute o áudio ao mesmo tempo