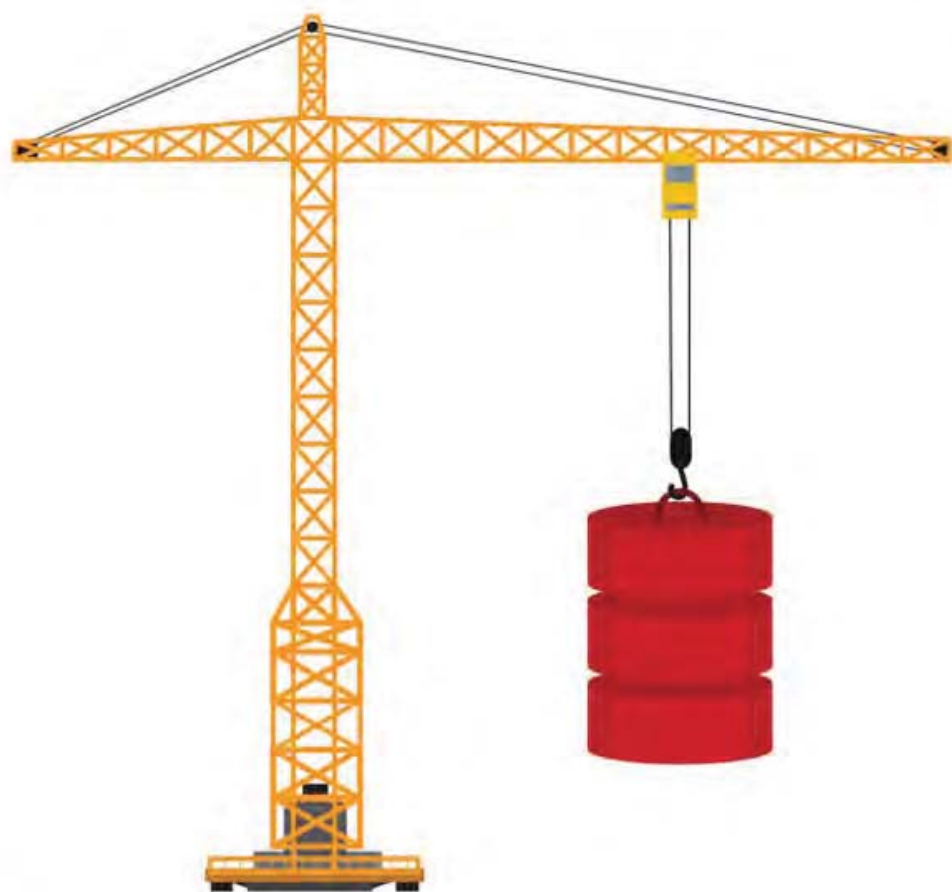


SQL

Uma abordagem para
banco de dados Oracle



CAPÍTULO 2

Banco de dados

Para se obter o armazenamento adequado da informação, é preciso que sejam escolhidos métodos que satisfaçam os requisitos de segurança e consistência. O banco de dados, principalmente o baseado no modelo relacional, é muito utilizado para este fim. Para tanto, deve-se entender suas características, seus conceitos e fundamentos. A utilização dos métodos de armazenamento através do uso de tabelas (linhas e colunas) garante seu armazenamento, bem como as regras definidas para o modelo relacional. A Integridade referencial precisa ser estudada e analisada. A álgebra relacional, que se trata de uma das técnicas para recuperação de dados pelos motores dos bancos de dados, deve ser entendida e operada de forma eficaz para alcançar o objetivo esperado. E por último, mas não menos importante, deve-se conhecer a linguagem padrão de acesso a dados, a SQL, para o modelo relacional, a qual reúne as abordagens das linguagens de definição, manipulação e controle dos dados.

2.1 INTRODUÇÃO AO BANCO DE DADOS RELACIONAL

Banco de dados relacional pode ser visto como uma coleção de dados. Esses dados estão disponibilizados de forma organizada e integrados, armazenados em forma de tabelas interligadas por chaves primárias e estrangeiras, constituindo uma representação de dados natural, podendo ser modificada sem restrições. Além disso, o banco pode ser utilizado por todas as aplicações relevantes sem duplicação de dados e sem a necessidade de serem definidos em programas, mas podendo ser, como composto de tabelas ou relações.

A abordagem relacional dos dados é baseada na observação de que arquivos podem ser considerados como relações matemáticas. Consequentemente, a teoria elementar de relações pode ser usada para lidar com vários problemas práticos que surgem com os dados desses arquivos. Com base em Edgar F. Codd, o principal expositor do modelo de dados relacional, pode-se dizer que um dos recursos poderosos do modelo relacional é a capacidade de permitir a manipulação de relações orientadas a conjuntos. Esta característica possibilitou o desenvolvimento de poderosas linguagens não procedurais com base na teoria dos conjuntos (álgebra relacional) ou em lógica (cálculo relacional). Em consequência disto, tabelas são tratadas como relações e as linhas dessas tabelas são usualmente conhecidas como tuplas, que conforme a maior parte da literatura pode ser definida como linha ou registro.

Um banco de dados relacional tem como objetivo a implementação do modelo de dados relacional, incorporando todas as características básicas como representação de entidades, atributos e relacionamentos. As entidades, também chamadas de tabelas, são constituídas por um conjunto de linhas não ordenadas, as tuplas. Cada linha que faz parte da tabela é constituída por um ou vários campos que são chamados de atributos da entidade. Vamos pegar como exemplo a tabela chamada EMP, do banco de dados que será usado em nosso treinamento:

The diagram shows a table with four columns: EMPNO, ENAME, JOB, and MGR. It contains 14 rows of employee data. Annotations include: a green callout for 'Nome do campo (Nome do atributo)' pointing to the EMPNO header; a blue callout for 'Linha (ou tupla)' pointing to the first row; a red callout for 'Valor do campo (valor do atributo)' pointing to the value 7369; and a yellow callout for 'Coluna (ou atributo)' pointing to the JOB column.

EMPNO	ENAME	JOB	MGR
7369	SMITH	CLERK	7902
7499	ALLEN	SALESMAN	7698
7521	WARD	SALESMAN	7698
7566	JONES	MANAGER	7839
7654	MARTIN	SALESMAN	7698
7698	BLAKE	MANAGER	7839
7782	CLARK	MANAGER	7839
7788	SCOTT	ANALYST	7566
7839	KING	PRESIDENT	
7844	TURNER	SALESMAN	7698
7876	ADAMS	CLERK	7788
7900	JAMES	CLERK	7698
7902	FORD	ANALYST	7566
7934	MILLER	CLERK	7782

Para que haja o relacionamento entre as entidades, mais precisamente entre as linhas de uma tabela, é preciso estabelecer o conceito de chaves. Os tipos de chaves geralmente encontradas num banco de dados relacional são três: chave primária (PK *Primary Key*), chave estrangeira (FK *Foreign Key*) e chave alternativa.

2.2 CHAVE PRIMÁRIA (ÍNDICE PRIMÁRIO)

Trata-se de um identificador único para tabela. Quando uma coluna ou combinações de coluna é instituída como chave primária, nenhum par de linhas da tabela pode conter o mesmo valor naquela coluna ou combinação de colunas. Uma chave primária deve seguir o conceito da minimalidade, isto é, que todas as suas colunas sejam efetivamente necessárias para garantir o requi-

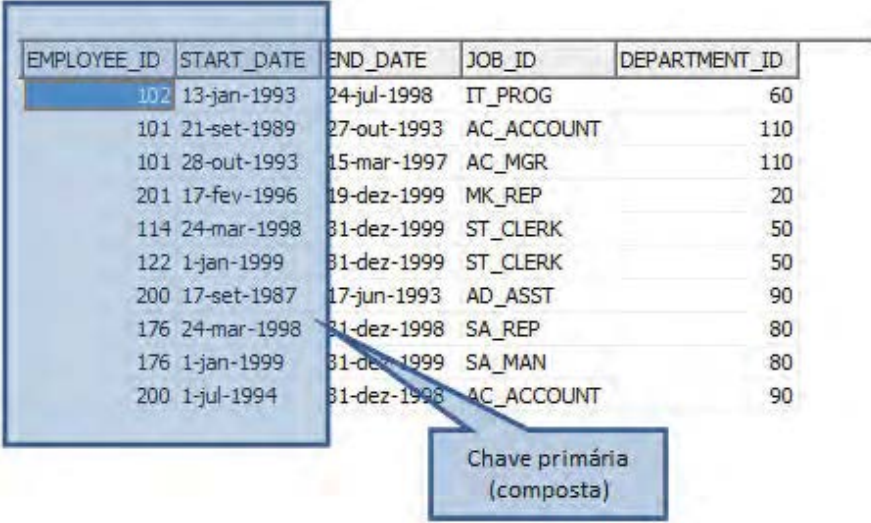
sito de unicidade de valores desta chave. Portanto, quando se consegue, com um atributo, garantir a unicidade de um registro, um segundo atributo compondo a chave é desnecessário. Quando se consegue com dois, um terceiro é desnecessário e assim por diante. Voltemos à tabela EMP:

EMPNO	ENAME	JOB	MGR
7369	SMITH	CLERK	7902
7499	ALLEN	SALESMAN	7698
7521	WARD	SALESMAN	7698
7566	JONES	MANAGER	7839
7654	MARTIN	SALESMAN	7698
7698	BLAKE	MANAGER	7839
7782	CLARK	MANAGER	7839
7788	SCOTT	ANALYST	7566
7839	ING	PRESIDENT	
7844	TURNER	SALESMAN	7698
7876	ADAMS	CLERK	7788
7900	JAMES	CLERK	7698
7902	FORD	ANALYST	7566
7934	MILLER	CLERK	7782



Chave primária (simples)

Fig. 2.2: No caso da tabela EMP, existe apenas uma coluna definida como chave primária (PK Simples). Portanto, a garantia da unicidade dos dados foi conseguida com apenas uma coluna da tabela, a coluna EMPNO.



EMPLOYEE_ID	START_DATE	END_DATE	JOB_ID	DEPARTMENT_ID
102	13-jan-1993	24-jul-1998	IT_PROG	60
101	21-set-1989	27-out-1993	AC_ACCOUNT	110
101	28-out-1993	15-mar-1997	AC_MGR	110
201	17-fev-1996	19-dez-1999	MK_REP	20
114	24-mar-1998	31-dez-1999	ST_CLERK	50
122	1-jan-1999	31-dez-1999	ST_CLERK	50
200	17-set-1987	17-jun-1993	AD_ASST	90
176	24-mar-1998	31-dez-1998	SA_REP	80
176	1-jan-1999	31-dez-1999	SA_MAN	80
200	1-jul-1994	31-dez-1998	AC_ACCOUNT	90

Chave primária
(composta)

Fig. 2.3: Neste caso foram necessárias mais de uma coluna (PK Composta) para garantir a unicidade dos dados, as colunas EMPLOYEE_ID e START_DATE.

2.3 CHAVE ESTRANGEIRA

Esta chave é determinada por uma coluna ou um conjunto de colunas possuidoras de um conjunto de valores válidos que estão presentes em outra tabela. Para enfatizar este conceito, pode-se dizer que uma chave estrangeira é uma coluna ou uma combinação de colunas, cujos valores aparecem necessariamente na chave primária de outra tabela. A chave estrangeira pode ser vista como sendo o mecanismo de implementação do relacionamento entre tabelas de um banco de dados relacional.

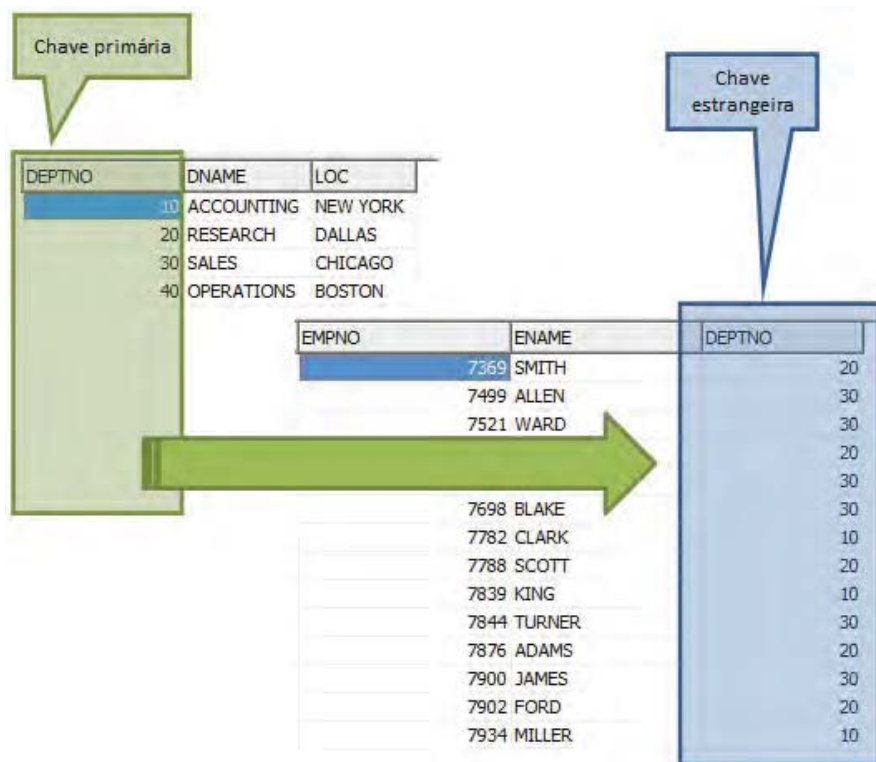


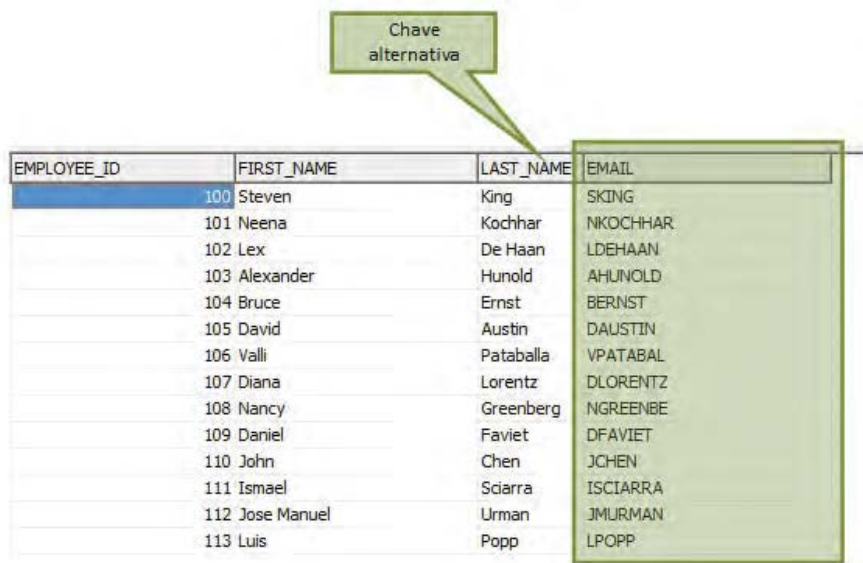
Fig. 2.4: As tabelas DEPT e EMP possuem suas colunas definidas como chave primária (EMPNO tabela EMP e DEPTNO tabela DEPT) e estão interligadas através da coluna DEPTNO, que existe tanto na tabela DEPT quanto na tabela EMP. Esta ligação garante que um empregado só poderá ser atribuído a um departamento existente na tabela de departamentos (DEPT).

Relacionamentos

As tabelas “conversam” entre si através de relacionamentos. Para que isso seja possível, é preciso que haja informações (colunas) que sejam comuns entre as tabelas. Um exemplo típico é a existência da coluna DEPTNO na tabela de empregados. A ligação ou relacionamento entre as duas tabelas é realizado através deste campo. Na figura anterior, existe um relacionamento entre as tabelas EMP e DEPT.

2.4 CHAVE ALTERNATIVA

Vimos que existem princípios para a criação de chaves primárias. O princípio da unicidade e o da minimalidade mostram que, embora se tenha várias colunas que podem fazer parte da chave primária (também chamadas de chaves candidatas), na maioria das vezes não se faz necessário, nem adequado, o uso delas para garantir a integridade. No momento da escolha de qual coluna deve ser usada, pode-se tomar como critério a escolha da coluna que se deseja usar nas chaves estrangeiras que referenciam a tabela em questão. Essas chaves candidatas podem ser utilizadas como índices secundários, úteis para acelerar operações de busca, embora nem sempre sejam eficazes. As chaves alternativas podem ser definidas como sendo um identificador único, inclusive sendo implementado como tal em uma base de dados.



The diagram shows a table with four columns: EMPLOYEE_ID, FIRST_NAME, LAST_NAME, and EMAIL. The first three columns are grouped by a bracket labeled 'Chave alternativa' (Alternative Key). The EMAIL column is highlighted with a green box, indicating it is the primary key. The table contains 13 rows of employee data.

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL
100	Steven	King	SKING
101	Neena	Kochhar	NKOCHHAR
102	Lex	De Haan	LDEHAAN
103	Alexander	Hunold	AHUNOLD
104	Bruce	Ernst	BERNST
105	David	Austin	DAUSTIN
106	Valli	Pataballa	VPATABAL
107	Diana	Lorentz	DLORENTZ
108	Nancy	Greenberg	NGREENBE
109	Daniel	Faviet	DFAVIET
110	John	Chen	JCHEN
111	Ismael	Sciarra	ISCIARRA
112	Jose Manuel	Urman	JMURMAN
113	Luis	Popp	LPOPP

Fig. 2.5: A tabela EMPLOYEES tem como chave primária a coluna EMPLOYEE_ID e como chave candidata a coluna EMAIL. A coluna EMAIL só conterá valores únicos, podendo ser utilizada como parâmetro em pesquisas realizadas nesta tabela.

Com relação ao uso de chaves estrangeiras, algumas regras devem ser observadas. A existência de uma chave estrangeira impõe restrições que devem ser garantidas ao executar diversas operações de alteração do banco de dados. Veja algumas a seguir:

- **Quando da inclusão de uma linha na tabela que contém a chave estrangeira:** deve ser garantido que o valor da chave estrangeira apareça na coluna da chave primária referenciada.
- **Quando a alteração do valor da chave estrangeira:** deve ser garantido que o novo valor de uma chave estrangeira apareça na coluna da chave primária referenciada.
- **Quando da exclusão de uma linha da tabela que contém a chave primária referenciada pela chave estrangeira:** deve ser garantido que na coluna chave estrangeira não apareça o valor da chave primária que está sendo excluída.
- **Quando da alteração do valor da chave primária referenciada pela chave estrangeira:** deve ser garantido que na coluna chave estrangeira não apareça o antigo valor da chave primária que está sendo alterada.

Outro aspecto que deve ser observado é a possibilidade de haver uma chave estrangeira que esteja referenciando a chave primária da própria tabela. Quando isso acontece, a solução se dá por meio do autorrelacionamento, isto é, a associação acontece na mesma tabela. A tabela-mãe e a tabela-filha são a mesma tabela. Vale ressaltar que uma chave estrangeira, sendo ela autorrelacionada ou não, pode conter valores nulos.

Chave estrangeira referencia a chave primária na mesma tabela

EMPNO	ENAME	JOB	MGR	HIREDATE
7369	SMITH	CLERK	7902	17-dez-1980
7499	ALLEN	SALESMAN	7698	20-fev-1981
7521	WARD	SALESMAN	7698	22-fev-1981
7566	JONES	MANAGER	7839	2-abr-1981
7654	MARTIN	SALESMAN	7698	28-set-1981
7698	BLAKE	MANAGER	7839	1-mai-1981
7782	CLARK	MANAGER	7839	9-jun-1981
7788	SCOTT	ANALYST	7566	9-dez-1982
7839	KING	PRESIDENT		17-nov-1981
7844	TURNER	SALESMAN	7698	8-set-1981
7876	ADAMS	CLERK	7788	12-jan-1983
7900	JAMES	CLERK	7698	3-dez-1981
7902	FORD	ANALYST	7566	3-dez-1981
7934	MILLER	CLERK	7782	23-jan-1982

Fig. 2.6: Todos os empregados da tabela EMP possuem um identificador (EMPNO). Entretanto, há outros que, além deste código, também possuem outro identificador que define quem é seu gerente. Neste exemplo, podemos perceber que o empregado ALLEN tem como gerente (MGR) o também empregado BLAKE. Neste nosso modelo, o autorrelacionamento acontece quando um empregado tem como gerente um empregado também cadastrado na mesma tabela, recebendo assim um código que existe na coluna EMPNO.

Quando se fala no uso de chaves, surge outro ponto importante, que é sobre o conceito de **regras de integridade**. O modelo relacional de banco de dados prevê duas regras gerais de integridade. São conhecidas como integridade de entidade e integridade referencial. De modo genérico, essas duas regras são aplicáveis a qualquer banco de dados que venha a implementar o modelo relacional.

Se os dados de um banco de dados estão íntegros, isso significa dizer que eles refletem corretamente a realidade representada pelo banco de dados e que são consistentes entre si. A integridade visa proteger os dados de um banco provendo sua segurança e integridade para que as informações sejam armazenadas de forma adequada. Com relação aos dois tipos de regras de integridade, pode-se dizer que:

2.5 INTEGRIDADE DE ENTIDADE

Um atributo, que faz parte de uma chave primária em uma relação, não poderá conter valores nulos, ou seja, ausência de informação. Outra característica importante da integridade de entidade diz respeito aos dados que são armazenados, pois os valores contidos nos campos de chave primária devem ser únicos, não havendo duplicidade de dados.

2.6 INTEGRIDADE REFERENCIAL

Esta integridade se trata de uma restrição que fixa que os valores que estão contidos em um campo de chave estrangeira devem, por sua vez, aparecer nos campos de chave primária da tabela referenciada. Sua principal função é garantir que as referências (ligações) entre as relações, relacionamentos entre tabelas, estejam compatíveis.

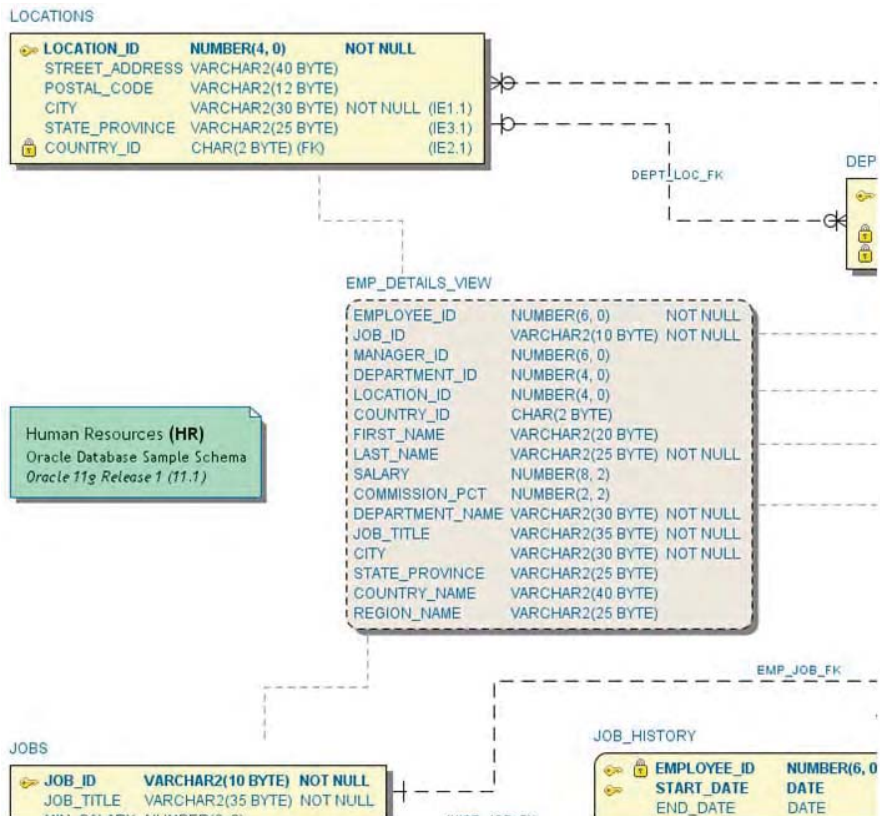


Fig. 2.7: A figura mostra com clareza as conexões existentes entre as tabelas LOCATIONS, JOBS, JOB_HISTORY, entre outras, através das chaves estrangeiras. Estas ligações garantem que os dados armazenados estejam consistentes e confiáveis. Trata-se de um exemplo, onde são mostradas as chaves primárias e estrangeiras existindo entre as tabelas.

Para que as regras de integridade sejam aplicadas de forma eficaz, é necessário que haja um sistema que as gerencie. Neste momento, entra em jogo o SGBD (Sistema de Gerenciamento de banco de dados). Um dos objetivos primordiais de um SGBD é a integridade de dados. Todas as regras de restrição existentes aplicáveis a um banco de dados são controladas por este sistema, ou seja, ele garante que as elas sejam aplicadas de forma a manter consistentes

os dados. De forma sucinta, pode-se dizer que o sistema de gerenciamento do banco de dados (também podendo ser chamado de DBMS) é o software que controla e manipula todos os acessos ao banco de dados. Alguns benefícios dos SGBDs são os seguintes:

- Simplicidade e uniformidade (o modelo relacional é compacto);
- Independência total dos dados (benefício mais importante);
- Interfaces de alto nível para usuários finais;
- Visões múltiplas de dados;
- Melhoria do diálogo entre o CPD e o usuário;
- Melhoria na segurança dos dados;
- Redução significativa do atravancamento de aplicações e do tempo;
- Gastos na manutenção;
- Alívio da carga de trabalho do CPD;
- Possibilidade de crescimento futuro e inclusão de novos dados devido à flexibilidade do sistema e independência de dados físicos.

Os SGBDs seguem também princípios que os caracterizam como um sistema de gerenciamento de bancos de dados relacionais. Em 1970, E. F. Codd, matemático da IBM já mencionado anteriormente, publicou um artigo intitulado como “Um modelo relacional de dados para grandes bancos de dados compartilhados”, contendo os princípios, ou regras, que definem um sistema de gerenciamento de bancos de dados como sendo um sistema relacional. São as que seguem:

- **Regra 1 :** todas as informações em um banco de dados relacional são representadas de forma explícita no nível lógico e exatamente em apenas uma forma, ou seja, por valores em tabelas.

- **Regra 2 :** cada um e qualquer valor individual (atômico) em um banco de dados relacional possui a garantia de ser logicamente acessado pela combinação do nome da tabela, no valor da chave primária e do nome da coluna.
- **Regra 3 :** valores nulos devem ser suportados de forma sistemática e independente do tipo de dados usado para representar informações inexistentes e informações inaplicáveis.
- **Regra 4 :** a descrição do banco de dados é representada no nível lógico da mesma forma que os dados ordinários, permitindo aos usuários autorizados que utilizem a mesma linguagem relacional aplicada aos dados regulares.
- **Regra 5 :** um sistema relacional pode suportar várias linguagens e várias formas de recuperação de informações, entretanto, deve haver pelo menos uma linguagem, como uma sintaxe bem definida e expressa por conjuntos de caracteres, que suporte de forma compreensiva todos os seguintes itens: definição de dados, definição de “views”, manipulação de dados (interativa e embutida em programas), restrições de integridade, autorizações e limites de transações (begin, commit e rollback).
- **Regra 6 :** todas as “views”, que são teoricamente atualizáveis, devem ser também atualizáveis pelo sistema.
- **Regra 7 :** a capacidade de manipular um conjunto de dados (relação) através de um simples comando deve-se entender as operações de inclusão, alteração ou exclusão de dados.
- **Regra 8 :** programas de aplicação permanecem logicamente inalterados quando ocorrem mudanças no método de acesso ou forma de armazenamento físico.
- **Regra 9 :** mudanças nas relações e nas “views” devem provocar o mínimo impacto possível nas aplicações.
- **Regra 10 :** as aplicações não podem ser afetadas quando ocorrem mudanças nas regras de restrições de integridade.

- **Regra 11 :** as aplicações não podem ser logicamente afetadas quando ocorrem mudanças geográficas nos dados.
- **Regra 12 :** se um sistema possui uma linguagem de baixo nível, essa linguagem não pode ser usada para subverter as regras de integridade e restrições definidas no nível mais alto.

Além destas regras, Codd também descreveu uma álgebra relacional fundamentando este sistema de gerenciamento de banco de dados relacional.