Tema 12

Tipos de datos nativos.

Tipos de datos y vistas para objetos.

Tipos de datos

- Almacenamiento y representación de datos.
- Cada tipo de datos tiene sus características y operaciones propias.
- Posibilidad de crear nuevos tipos de datos a partir de los propios del SBGD.

Cadenas de caracteres

- ¿Qué tamaño ocupan?
- ¿Qué caracteres queremos representar?
- ¡Cuidado con la codificación de caracteres!
- Tipos de datos:
  - CHAR
  - VARCHAR y VARCHAR2
  - NCHAR y NVARCHAR2
  - CLOB y NCLOB

Cadenas de caracteres

- CHAR:
  - Tamaño fijo entre 1 y 2000 bytes.
  - Siempre ocupan lo mismo en disco.
  - Se rellena con espacios si no llega al tamaño del campo.
- VARCHAR y VARCHAR2:
  - Preferible usar VARCHAR2.
  - Tamaño variable entre 1 y 4000 bytes.

Semántica de bytes/caracteres

- Un carácter puede ocupar varios bytes.
  - Depende de la codificación de caracteres y de qué carácter sea.
- Podemos declarar campos para ocupar un cierto número de bytes o de caracteres.
- 30 caracteres UTF-8 = 90 bytes.
- Preferible no especificar la semántica.
- NLS_LENGTH_SEMANTICS.

Cadenas de caracteres

- NCHAR y NVARCHAR2.
  - Casi idénticos a CHAR y VARCHAR2.
  - Solo aceptan codificación UTF-8 y UTF-16.
  - Tres bytes por carácter.
  - Muy útil para bases de datos en varios países.
- CLOB y NCLOB.
- LONG.
  - No se debe usar.
    - está solo por compatibilidad.

Tipos de datos numéricos

- NUMBER.
  - 38 dígitos de precisión decimal (enteros y reales).
  - Precisión y escala.
  - Almacenamiento en notación científica:
    - Mantisa y exponente enteros.
- BINARY_FLOAT y BINARY_DOUBLE.
  - Aritmética binaria:
    - Cálculos más rápidos.
  - Ocupan 5 y 9 bytes respectivamente.

Fecha y hora

- DATE.
  - Almacena siglo, año, mes, día, hora, minuto y segundo: siete bytes.
  - Formato de fecha: NLS_DATE_FORMAT.
    - Funciones TO_DATE y TO_CHAR para otros formatos.
  - Soporta el calendario juliano.
  - Permite comparación de fechas, resta de fechas y sumar o restar constantes.

Fecha y hora

- TIMESTAMP [n].
  - Idéntico a DATE, pero almacena fracciones de segundo (hasta 1/1 000 000 000 segundos).
- TIMESTAMP [n] WITH TIME ZONE.
  - Incluye siempre una zona horaria.
    - Por ejemplo, 18/12/2011 14:16:43 UTC+1.
- TIMESTAMP [n] WITH LOCAL TIME ZONE.
  - Siempre se muestra con la zona horaria del servidor.

RAW y LONG RAW

- Datos que no pueden ser interpretados por Oracle.
  - gráficos, documentos, sonidos…
- RAW: longitud variable hasta 2000 bytes.
  - Soporta indexación.
- LONG RAW: longitud variable hasta 2 GB.
  - No soporta indexación.
- En general, no realiza conversión de caracteres.
- Suministrados por compatibilidad con versiones anteriores.
  - Oracle recomienda usar BLOB y BFILE.

ROWID y UROWID

- Usados internamente por Oracle.
- ROWID almacena direcciones físicas o lógicas de filas de la base de datos.
- UROWID permite además almacenar ROWID de bases de datos que no sean Oracle.
- No forman parte de los datos de la base de datos, salvo que se definan columnas de dichos tipos.

ROWID físicos

- Contienen la dirección física de una fila.
- Podemos usarlos para ver la organización de una tabla.
- Formatos:
  - Extendido:
    - Especifica el número de objeto de datos, el identificador de archivo, el bloque de datos y la fila.
  - Restringido:
    - Especifica el bloque de datos, la fila y el identificador de archivo.
    - Ya no se usa.
      - Está solo por compatibilidad con versiones antiguas.

ROWID lógicos

- Se basan en la clave primaria de las OIT (éstas no tienen dirección física permanente).
- Oracle los usa para la construcción de índices secundarios.
- Pueden incluir una pista o suposición física que indica dónde estaba el bloque de la fila al crearse el índice secundario.
- No se actualizan al mover una fila; hay que reconstruir el índice secundario.
- Estadísticas mediante DBMS_STATS y ANALYZE.

Comparación ROWID físicos y lógicos

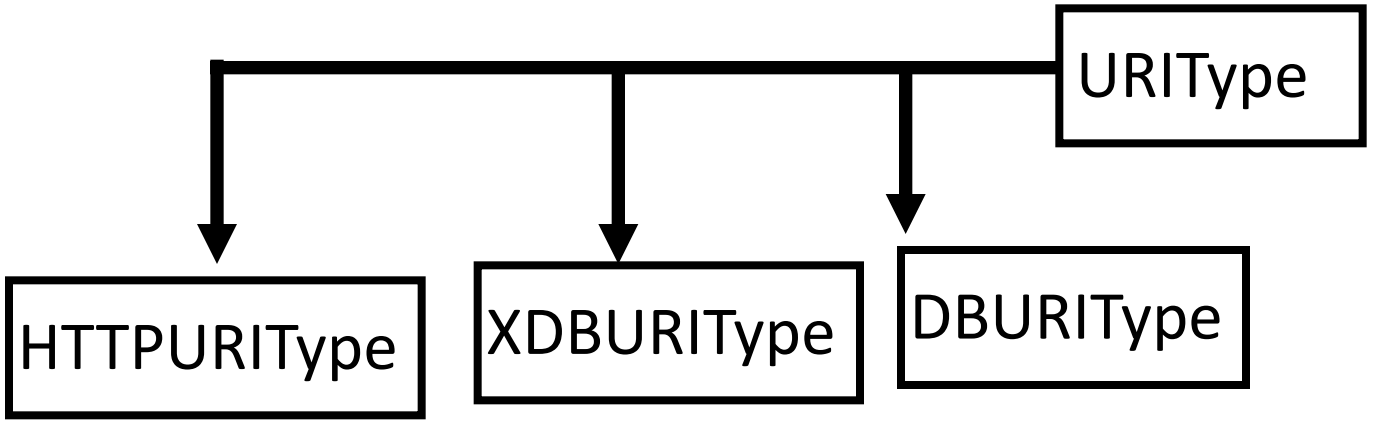
- Ambos se obtienen con la pseudocolumna ROWID.
- Son la forma más rápida de acceder a una fila.
  - En los lógicos puede implicar leer más de un bloque.
- Los ROWID lógicos cambian al modificar su clave primaria asociada, y los físicos, permanecen inmutables a los cambios de su fila asociada.
- Los ROWID lógicos no permiten obtener información de la organización de una tabla.

Extensible Markup Language (XML)

- Formato estándar desarrollado por W3C (World Wide Web Consortium).
- Permite manipular información XML almacenada en la base de datos (insertar, extraer, preguntar).
- Se trata como cualquier otro tipo de dato integrado o definido por el usuario.
  - Puede ser un tipo de dato de una columna o tabla o un parámetro de E/S de un procedimiento PL/SQL.

Identificador universal de recursos (URI)

- El supertipo URI es un tipo abstracto que ofrece un conjunto de funciones para obtener los valores que son apuntados (referenciados) por los URI.
- La implementación de dichas funciones tienen que ser definidas por los subtipos de URIType.



Funciones del supertipo URI

- GETBLOB → Devuelve el tipo BLOB encontrado en la dirección especificado por la URL.
- GETCLOB → Devuelve el tipo CLOB encontrado en la dirección especificada por la URL.

GETCONTENTTYPE
GETEXTERNALURL
GETURL

} Devuelve la URL almacenada en la instancia de tipo URIType.

GETXML → Devuelve el tipo XMLType encontrado en la dirección especificada por la URL.

Multimedia

- Se encuentra en el esquema ORDSYS.

- Es el tipo de dato que Oracle ofrece para poder manipular información multimedia.
  - ORDAudio (para datos de sonido).
  - ORDImage (para datos de imagen).
  - ORDImageSignature (para compactar la representación del color, textura y forma de la información de datos de imagen).
  - ORDVideo (para datos de vídeo).
  - ORDDoc (para cualquier dato multimedia).

Objeto

- Constituyen todos los tipos de dato integrados y los definidos por usuarios.

- Se trata de los bloques base para los tipos de dato que modelan la estructura y comportamiento de los datos en las aplicaciones.

- Nombre
  - Lo identifica de forma unívoca dentro del esquema
- Atributos
  - Modelan la estructura de la entidad equivalente en el mundo real
- Métodos
  - Implementan las operaciones que las aplicaciones pueden emplear en las entidades del mundo real

Métodos

- Objetivo:
  - Hacer posible el acceso a los datos de un objeto
- Miembro:
  - Permiten a la aplicación acceder a los datos de la instancia de un objeto
- Estático:
  - Se pueden invocar cualificando el método por el nombre de su tipo TYPE_NAME.METHOD
- Métodos de comparación:
  - Para comparar instancias de objetos

Métodos de comparación (Comparison)

- Método de mapas.
  - Se basan en la capacidad que tiene Oracle de comparar tipos de datos integrados.

- Método de ordenación.
  - Se establece una relación de comparación entre dos objetos de un tipo de objeto determinado.
  - El resultado de la comparación se representa devolviendo un número específico.

Conversión de datos

- Por lo general en una misma expresión no se pueden mezclar valores de diferentes tipos de datos.

  5 + 8 + ‘D’ = ¿?

- Conversión implícita.
  - El usuario supone que la conversión automática hará lo que él espera de ella.
  - Es impredecible.

- Conversión explícita
  - El usuario sabe lo que la conversión hará.
  - Es predecible.

Tablas de objetos

- Tipo especial de tabla para almacenar objetos.
- Varias maneras de visualizarla:
  - Con una única columna donde cada entrada es un objeto (1).
  - Con múltiples columnas para representar también los atributos de los objetos (2).

(1)

Objeto 1
Objeto 2
Objeto 3
Objeto 4
Objeto 5

(2)

Objeto 1	Atributo 1	Atributo 2	Atributo 3
Objeto 2	Atributo 1	Atributo 2	Atributo 3
Objeto 3	Atributo 1	Atributo 2	Atributo 3
Objeto 4	Atributo 1	Atributo 2	Atributo 3
Objeto 5	Atributo 1	Atributo 2	Atributo 3

Tablas de objetos

- Cada objeto de la tabla tiene un identificador que una de dos:
  - Ha sido generado automáticamente por Oracle por defecto.
  - Se basa en la clave primaria del objeto.
- El identificador se almacenará en una columna oculta de la tabla.

Tipo colección

- Son unidades de datos compuestas de un número indefinido de elementos del mismo tipo.

- Existen tres tipos.
  - VARRAY.
  - Arrays asociativos.
  - Tablas anidadas.

VARRAY

- Conjunto de elementos ordenados donde todos los elementos son del mismo tipo.
- Denominados Varray porque pueden ser variables (Variable Array).
- Cada elemento tiene asociado un índice que indica su posición dentro del array.

Tablas anidadas

- Son conjuntos de elementos no ordenados, donde todos los elementos son del mismo tipo de datos.

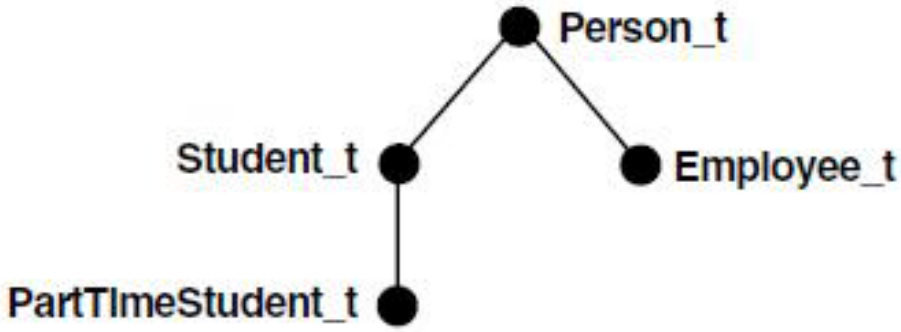
- Al igual que las tablas de objetos, pueden ser visualizadas de dos maneras: con una única columna o con varias si almacenan objetos.

Herencia

- Se utiliza para representar subtipos.
- Cuando un objeto hereda de otro, es su subtipo, y además tomará todos los atributos y métodos del supertipo.
- No se permite herencia múltiple.
- Se pueden sobrescribir o añadir métodos heredados.

Herencia

Figure 27–1 A Type Hierarchy



Características de tipos

- Finales.
- No finales.

- Instanciables.
- No instanciables.

```
CREATE TYPE Address_t AS OBJECT(...) NOT INSTANTIABLE NOT FINAL;
CREATE TYPE USAddress_t UNDER Address_t(...);
CREATE TYPE IntlAddress_t UNDER Address_t(...);
```

Funciones definidas por usuarios

- Un usuario puede crear una nueva función y proporcionar un conjunto de rutinas.
- Estas funciones pueden ser utilizadas de manera similar a las funciones por defecto.
- Otorga una gran flexibilidad para trabajar con datos complejos, como imágenes o vídeo.

Evolución de los tipos de datos

- Dependencias.
- Elementos inválidos.
- Revalidación.
- Alteración de tipos dependientes.
- Propagación de cambios de tipos.

Vistas de objeto

- Una vista de objeto es una tabla virtual de objetos.
- De manera análoga, una vista de objeto es un objeto virtual (tomado de una tabla de objetos).
- Podemos crear tablas de objetos virtuales a partir de los datos almacenados en columnas virtuales.

Funciones y ventajas de las vistas de objeto

- Proporcionan la habilidad de ofrecer acceso especializado o restringido a los datos y a los objetos de la base de datos.
- Pueden favorecer el rendimiento.
- Ofrecen la flexibilidad de ver los mismos datos de diversas maneras.

Uso de las vistas de objeto

- Se puede hacer referencia a las vistas de objetos en sentencias SQL.
- Se puede acceder a los datos de las vistas de objetos utilizando las mismas llamadas que usamos para los objetos de las tablas de objetos.

Actualización de las vistas de objeto

- Se puede actualizar, insertar y suprimir datos en una vista de objeto utilizando el mismo lenguaje de manipulación de datos que utilizamos para las tablas de objetos.
- Existen algunas restricciones.
  - Se solucionan con disparadores (INSTEADOF).

Jerarquía de vistas

- Una vista de objeto puede crearse como una subvista de otra vista de objeto.
- Es posible una jerarquía de vistas.

Figure 27–2 Multiple View Hierarchies