Week 4 Homework 2

Matt Rieser

SDEV 350 – Section 7980

November 11, 2018

Professor Ivan De Los Santos

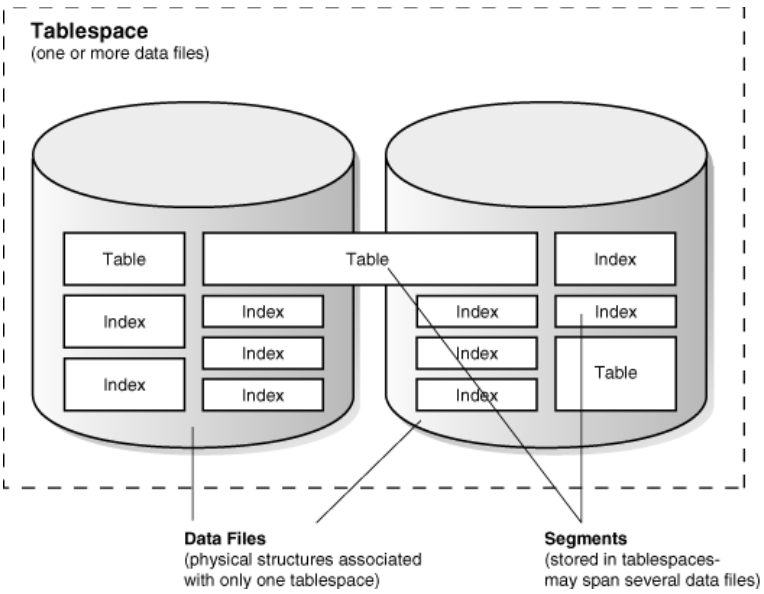
Oracle 12c database

A relational database management system

Physical structure of a RDBMS

Included are data, temp, control, and online redo log files.

Data file – a physical file on disk that was created by Oracle Database, contains tables and indexes. Tablespace data is stored in data files (shown below).



Temp file – a data file that belongs to a temporary tablespace. Data is written in a proprietary format created by oracle.

Data and temp files can either be online or offline. Tablespaces share this trait too.

Control file – a root file that track the physical component of the database. Only one associated with each database.

Information included within each are the database name/ database unique identifier, time stamp of database creation, information about data files/online redo log files/archived redo log files, tablespace information, and RMAN backups

Online redo log – a set of files containing record of changes made to data

This is used to protect against data loss. Stores all changes made to a data including undos. Helpful in a instance failure to retrieve uncommitted transactions, and schema and object management statement

Oracle has several options for storing and managing the previously explained files.

These include:

- Oracle Automatic Storage Management
- Operating system file system
- Cluster file system

Logical structure of a RDBMS

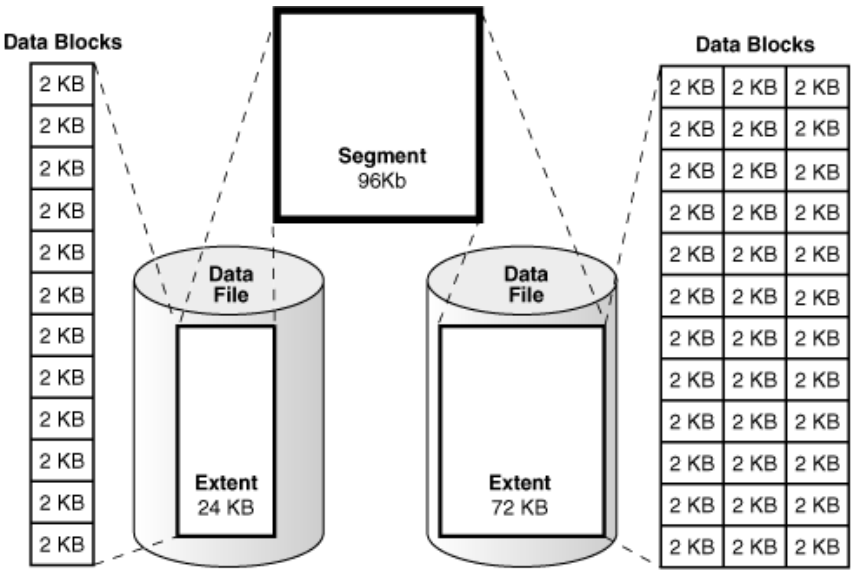
Included are data blocks, extents, segments, and tablespaces.

Data block – is the smallest logical unit of data storage

Extent – a set of logically contiguous data blocks allocated for storing a specific type of information

Segment – a set of extents allocated for a specific database object, such as a table.

Tablespace – a database storage unit that contains one or more segments



Above - The relationship between data blocks, extents, segments, and tablespaces. It is showing a segment that has two extents stored in different data files.

Oracle manages logical space depending on which tablespace you choose

- Locally managed tablespaces – uses bitmaps in tablespaces to manage extents
- Dictionary managed tablespace – uses the data dictionary to manage extents

Data Dictionary – read-only collection of database tables and views containing reference information about the database, its structures, and its users.

Memory architecture of a RDBMS

After an instance is started, a memory area is allocated for it

The memory area stores information such as the following:

- Program code
- Information about each connected session, even if it is not currently active

- Information needed during program execution, for example, the current state of a query from which rows are being fetched
- Information such as lock data that is shared and communicated among processes
- Cached data, such as data blocks and redo records, that also exists on disk

Memory structures included

- System global area (SGA) – a group of shared memory structures that contain data and control information exclusively for oracle processes
- Program global area (PGA) – nonshared memory region that contains data and control information exclusively for oracle processes
- User global area (UGA) – memory associated with a user session
- Software code areas – portions of memory used to store code that is running or can be run

Memory management options include

- Automatic memory management – specify target size for database instance memory and it will automatically tune memory as needed between the SGA and PGA
- Automatic shared memory management – specify a target size for SGA and have the option of setting an aggregate of target size for the PGA
- Manual memory management – set many initialization parameters to manage the SGA and PGA, instead of setting just the total memory and having it automatically distribute

Processes architecture of a RDBMS

Oracle users must run the following in order to access an instance:

- Application or oracle database utility – precompiler program or database tool like SQL*Plus or SQL Developer
- Oracle database code – each user has oracle code that interprets and processes each the SQL statements (provided the user)

Types of Processes

- Client process – runs the application or database code
- Oracle processes – background processes, server processes, and slave processes

Application architecture of a RDBMS

Client/server architecture:

- client – runs a database application like SQL *Plus, Visual Basic, or a SQL Developer data entry program. This program accesses the database information and interacts with user inputs and commands
- server – runs the Oracle data software and handles the functions required for concurrent, shared data access to an Oracle database.

Benefits to Client/server –

- Client applications are not responsible for performing data processing
- Client applications are not dependent on the physical location of the data

- Oracle Database exploits the multitasking and shared-memory facilities of its underlying operating system
- Client workstations or terminals can e optimized for presentation for data, while the server can be optimized for processing and storage of data
- Networked environments can use inexpensive client machines to access remote data in the server
- Networked environments shared data is stored on the servers, making it easy and efficient to manage concurrent access
- Network traffic is minimized due to requests and results being the only things that are sent

Network architecture of a RDBMS

Oracle Net Services - a collection of networking components that provide enterprise-wide connectivity solution in distributed , heterogeneous computing environment

How it works?

Oracle Database protocols accept SQL statements from the interface of the Oracle application and then package them for transmission to Oracle Database (through API or higher level protocols)

Google Cloud datastore

A NoSQL document database built for automatic scaling, high performance, and application development.

Physical

The physical aspects of storage are all handle through Google's cloud storage

Storage calculations are handled as such:

String size - calculated as the number of UTF-8 encoded bytes + 1 and keys, kind names, namespace names (default size is 0), property names, and string property values.

Example – The name of the **description** property uses 11 bytes + 1 byte, for a total of 12 bytes.

Key size –

- The namespace string size (if not in the default namespace)
- The full key path string size (integer IDs are 8 bytes each)
- 16 bytes

Property size –

- The property name string size
- The property value's size

Entity size –

- The key size
- The sum of the property sizes

- 32 bytes

Index entry size – calculated by built – in and composite indexes

Built in indexes

- The key size of the indexed entity
- The sum of the indexed property names
- The sum of the indexed property values
- The size of the indexed entity kind name
- 32 bytes

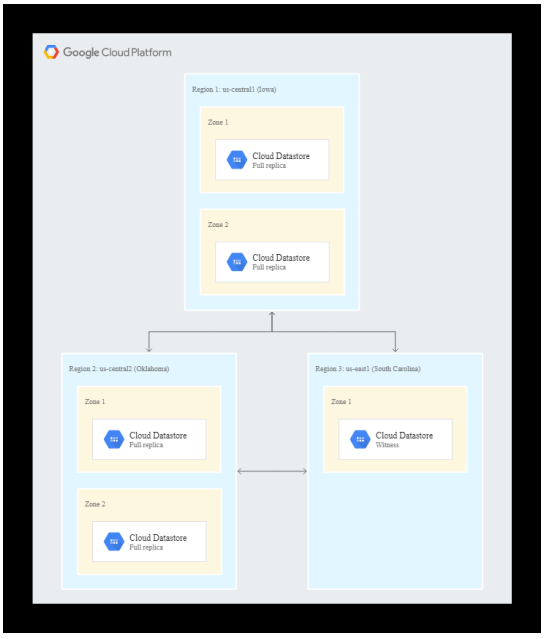
Composite indexes

- The key size of the indexed entity
- The sum of the indexed property values
- 32 bytes

Redundancy

Cloud Datastore offer two levels of redundancy

- Regional replication – data is replicated in at least three different zones within the same region. Resilient to zonal outages and best for achieving low write latency. Tip - might want to co-locate your application's compute machines in the same region.



- Multi-region replication – data is replicated in multiple zones across at least two different regions. Increases availability and redundancy but has higher latency. (pictured to the left)

Cloud Datastore supports data exports to Cloud Storage for archival or disaster recovery purposes

Logical

How a NoSQL Cloud Database differs from a RDBMS ?

Concept	**Cloud Datastore**	**Relational database**
Category of object	Kind	Table
One object	Entity	Row
Individual data for an object	Property	Column
Unique ID for an object	Key	Primary Key

Google Cloud datastore scales automatically to massive data sets. This provides higher performance when dealing with more traffic to the database.

How it differs from traditional databases like relational, MySQL, PostgreSQL, etc. –

- Writes the scale of the data automatically by distributing data as necessary
- Reads scale because the only queries supported are those whose performance scales with the size of the result set (as opposed to the data set).
- Types of queries that can be executed are more restrictive than those allowed on a relational database with SQL. For example Cloud Datastore does not include support for

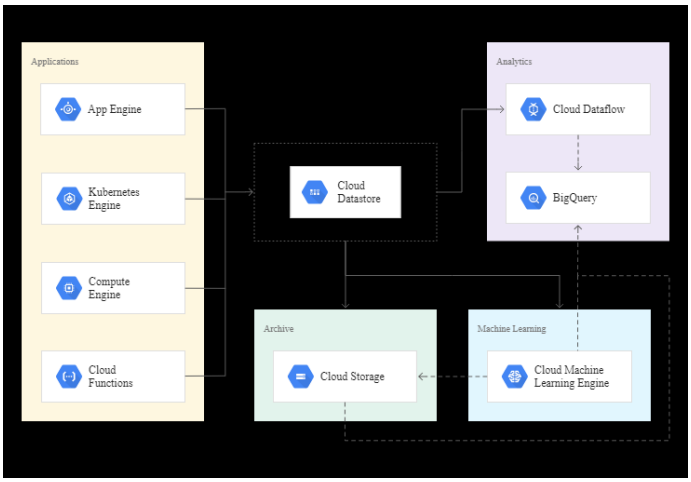
join operations, inequality filtering on multiple properties, or filtering on data based on results of a subquery.

- Doesn't require a schema or require entities of the same kind to have a consistent set of properties (but there is an option to do so if desired)

Process and Application Architecture

On the Google Cloud platform there are several different services for hosting applications to access the Google Cloud datastore.

Google App Engine – is a platform as a service (PaaS) that allows you to deploy applications directly into a managed, auto-scaling environment.



(Left is an example of Cloud Datastore implementation with other Google services)

Google Kubernetes Engine – provides finer-grained control for container-based applications, allowing you to easily orchestrate and deploy containers across a cluster.

Google Compute Engine – provides high-performance virtual machines for maximum flexibility.

Cloud Functions Engine – enables you to deploy event-driven microservices.

The reason I chose Google cloud database to compare with Oracle 12c was because it is a newer platform and concept. Being that Google datastore is No SQL and Oracle 12c is based around SQL, it seemed like an interesting dynamic. The dynamic being the differences between having SQL and not having SQL in a database architecture. Another was that Google database is a cloud-based service verse Oracle 12c which more client/server based.

Resources

Oracle 12c resources-

(2006, December 08). Retrieved November 19, 2018, from https://docs.oracle.com/database/121/nav/portal_4.htm

Google datastore resources-

Architecture: Complex Event Processing | Architectures | Google Cloud. (n.d.). Retrieved November 19, 2018, from https://cloud.google.com/solutions/architecture/complex-event-processing

Building Scalable Web Applications with Cloud Datastore | Architectures | Google Cloud. (n.d.). Retrieved November 19, 2018, from https://cloud.google.com/solutions/building-scalable-web-apps-with-cloud-datastore

Database Concepts. (2017, July 10). Retrieved November 19, 2018, from https://docs.oracle.com/database/121/CNCPT/intro.htm#CNCPT001