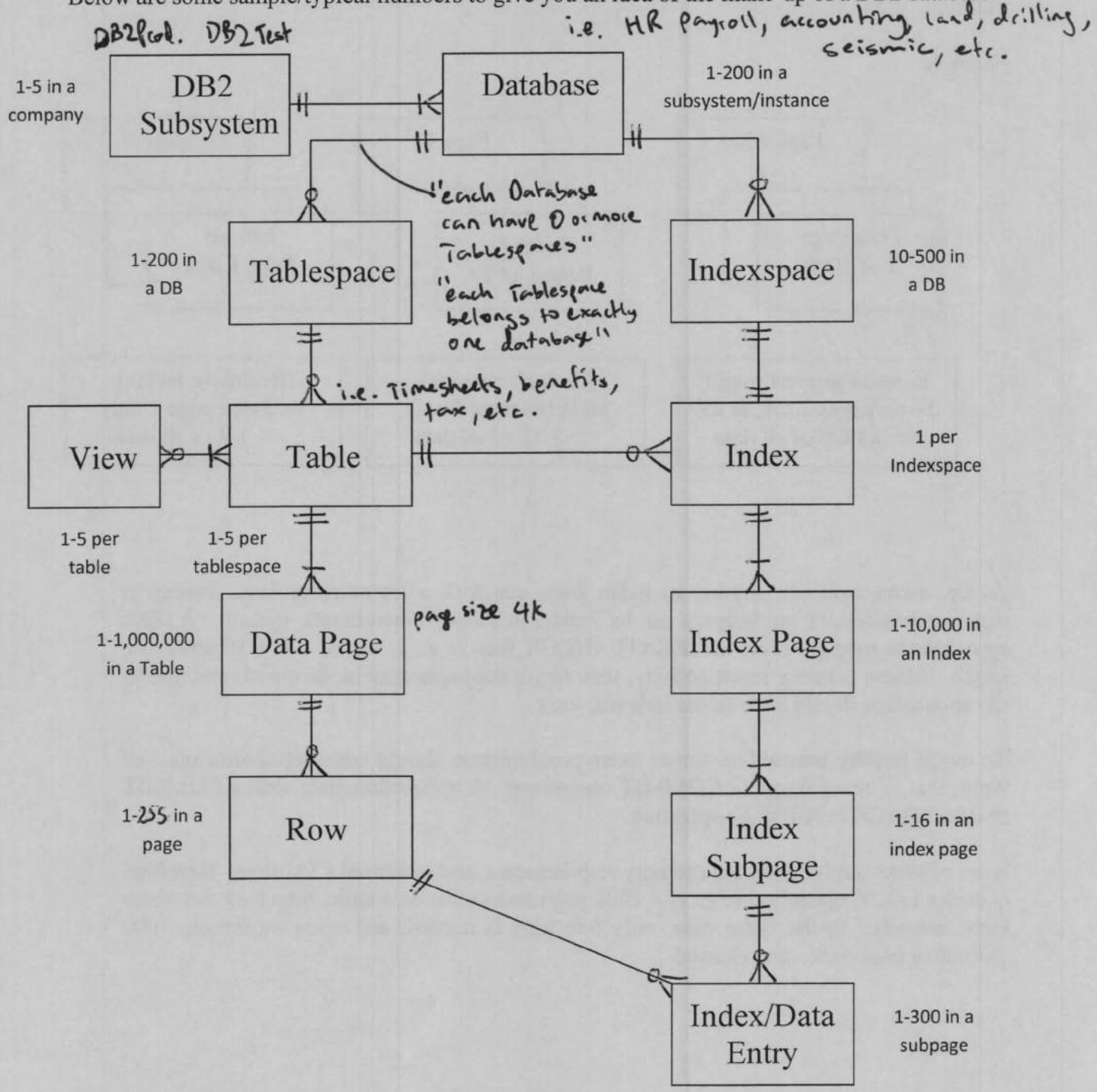
404 using crows Foot Notation

0 or many
1 or many
exactly one
zero or one

Case Study: DB2 for z/OS

Here are some important DB2 objects in an enterprise-scale (large) IBM environment, shown in Crow's Foot notation. A company may have several DBMSs, and may run multiple instances of each. Physical objects (files) are *tablespaces* and *indexspaces*; they contain pages. A tablespace is a unit of backup and recovery. In some shops, there is often only one table per tablespace. A view can involve one or more tables. Most views are not materialized (i.e., not stored on disk). Below are some sample/typical numbers to give you an idea of the make-up of a DB2 database.

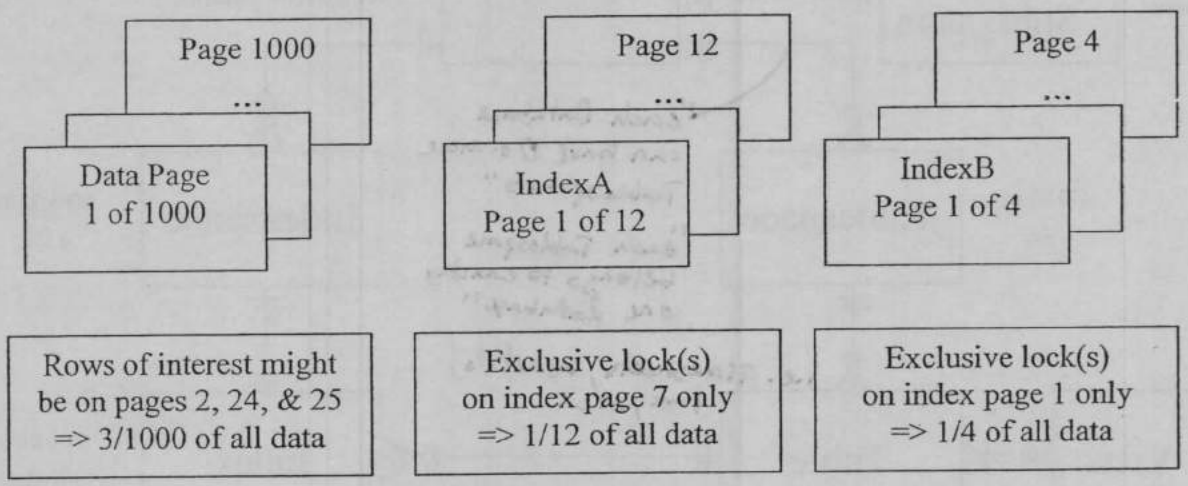


Deadlocks occur infrequently. DB2 checks for deadlocks every 15 seconds, although a Systems Programmer/Administrator (SYSADM) can change this installation default.

DB2 resolves deadlocks by aborting ("rolling back") the transaction involved in the deadlock that has the fewest log records.

Note that SQL INSERTs, UPDATEs, and DELETEs may require the locking of multiple indexes.

Example:



As the above example shows, an index page can lock a lot of rows (e.g., during an insert/update/delete); so, indexes can be locked in subpage increments, instead. A DBA specifies the subpage level at CREATE INDEX time (e.g., 1, ½, ¼, 1/8, 1/16 units of a page). If there is heavy insert activity, then larger subpages may be favoured (and maybe the application should have fewer indexes, too).

To avoid lengthy transaction times, users/programmers should consider shorter units of work, that is, more frequent COMMIT operations. A transaction ends with a COMMIT or ROLLBACK (=ABORT) operation.

A lot of locks imply increased memory requirements, and increased CPU time; therefore, consider LOCK escalation (e.g., if > 1000 page locks exist on a table, then lock the whole table instead). In the latter case, only one lock is needed, and upon escalation, 1000 individual page locks are released.