How can we store large-object types?

Все большие объекты хранятся в одной системной таблице с именем  pg\_largeobject. Для каждого большого объекта также имеется запись в системной таблице  pg\_largeobject\_metadata. Большие объекты можно создавать, изменять и удалять, используя API чтения/записи, подобный стандартному API для работы с файлами. (max – 4TB)

PostgreSQL также поддерживает систему хранения, названную «TOAST», которая автоматически переносит значения, не умещающиеся в одну страницу таблицы, в дополнительную область хранилища. (max – 1GB)

Реализация разбивает большие объекты на чанки chunks и сохраняет эти фрагменты в строках таблицы. Быстрый поиск нужного фрагмента обеспечивается индексом B-дерева в этой таблице. Чанки больших объектов могут быть непоследовательными. (если приложение откроет новый большой объект, переместится к смещению 1000000 байт и запишет несколько байт, это не приведёт к выделению лишнего 1000000 байт в хранилище; записаны будут только фрагменты, покрывающие диапазон собственно записанных байт. Операция чтения, однако, прочитает нули для всех неразмещённых в хранилище байт, предшествующих последнему записанному фрагменту).

Клиентские приложения, которые используют эти функции, должны включать заголовочный файл libpq/libpq-fs.h и компоноваться с библиотекой libpq.

Создание большого объекта

inv\_oid = lo\_create(conn, desired\_oid);

inv\_oid = lo\_creat(conn, INV\_READ|INV\_WRITE);

Импорт большого объекта

Oid lo\_import\_with\_oid(PGconn \*conn, const char \*filename, Oid lobjId);

Экспорт большого объекта

int lo\_export(PGconn \*conn, Oid lobjId, const char \*filename);

Открытие существующего большого объекта

inv\_fd = lo\_open(conn, inv\_oid, INV\_READ|INV\_WRITE);

Запись данных в большой объект

int lo\_write(PGconn \*conn, int fd, const char \*buf, size\_t len);

Чтение данных из большого объекта

int lo\_read(PGconn \*conn, int fd, char \*buf, size\_t len);

Усечение большого объекта

int lo\_truncate(PGcon \*conn, int fd, size\_t len);

Удаление большого объекта

int lo\_unlink(PGconn \*conn, Oid lobjId);

What is the difference between privilege, role and user?

В PostgreSQL user это роль с правом входа. Привилегии это возможность взаимодействовать с объектами для пользователей, которые не являются superuser или владельцом объекта(они имеют эти привилегии по умолчанию, но владельца можно ограничить). Чтобы дать возможность взаимодействовать с объектами используется GRANT, а отозваны REVOKE.