1)

a)  
Duplicate records  
not fully normalize  
bi)  
STAFF NAME  
STAFF postal\_coed  
ii)name  
iii) unnormalised form  
iv)  
STAFF=(Name)  
ADDRESS = (street, city, state, postal\_code)  
c)  
INVOICE (HA\_hospital, DATE, NAME, ADDRESS, PET\_NAME, PROCEDRE, AMOUNT)

2)

A)

SELECT COUNT( DateAdmitted)  
FROM Animals   
WHERE DateAdmitted = ’18 April 2015’);

B)  
SELECT DISTINCT TYPE  
FROM Animals  
UNION  
SELECT AdoptDate  
FROM Adoption  
WHERE AdoptDate = “14 June 2015”;  
C)  
SELECT DISTINCT TYPE  
FROM Animals  
UNION   
SELECT AdoptDate  
FROM Adoption  
WHERE AdoptDate IS NULL  
ORDER BY TYPE;  
D)  
SELECT Name, Address  
FROM Adopter  
WHERE COUNT(SIN) > 2;

* 3)  
  A)  
  **Authentication** is the process of verifying who you are. When you log on to a PC with a user name and password you are authenticating.
* **Authorization** is the process of verifying that you have access to something. Gaining access to a resource (e.g. directory on a hard disk) because the permissions configured on it allow you access is authorization.

B)   
Reduced privilege administration  
Rather than granting the same set of privileges explicitly to several users, you can grant the privileges for a group of related users to a role, and then only the role must be granted to each member of the group.  
Dynamic privilege management

If the privileges of a group must change, then only the privileges of the role need to be modified. The security domains of all users granted the group's role automatically reflect the changes made to the role.

Selective availability of privileges

You can selectively enable or disable the roles granted to a user. This allows specific control of a user's privileges in any given situation.

Application awareness

The data dictionary records which roles exist, so you can design applications to query the dictionary and automatically enable (or disable) selective roles when a user attempts to execute the application by way of a given user name.

Application-specific security

You can protect role use with a password. Applications can be created specifically to enable a role when supplied the correct password. Users cannot enable the role if they do not know the password.  
C)  
LOGIN : SYSTEM  
PASSWORD : UTAR  
CREATE USER FES\_PROFILE PROFILE “DEFAULT” IDENTIFIED BY FES;  
CREATE USER FES\_PROFILE LIMIT  
password\_grace\_time 30 (change password every 30 days)  
CONNECT\_TIME 300 (overall connection time 5h)  
failed\_login\_attempts 5 (limit failed login attempt to 5 in a row)  
 password\_lock\_time 2  
password\_life\_time 3  
password\_reuse\_time 30  
password\_reuse\_max 10;  
ALTER USER SCOTT\_PROFILE PROFILE DIECS;

4)

a) σAmount > 2000 (Expense)  
b) π SSN σTo\_city = ‘Chandigarh’ (SALESPERSON X TRIP)  
C) π Amount σSSN = ‘234-56-7890’ (SALESPERSON X EXPENSE)  
  
Data Complexity

Data in an RDBMS resides in multiple tables, which are linked to each other through shared key values. An RDBMS does not force database designers to impose a coherent table structure; inexperienced programmers may design systems that create unnecessary complexity or limit the future development of the database through poorly chosen data types. The flexibility of an RDBMS presents a double-edged sword. Experienced designers work magic, but inexperienced designers wreak havoc on a company's data.

Broken Keys and Records

Relational databases require shared keys to link information spread across several tables. For example, a customer table may include client demographics, with a unique index number identifying the record within the table. A sales table may identify the customer only by that index number. If the data types linking the keys are different, the database cannot link the records without additional rework by the report developer. Likewise, if a table lacks a unique key, the database may return inaccurate results. If the application accessing a database isn't coded to lock records during an edit, users could inadvertently corrupt data, leading to broken records.  
  
Developer Expertise

As the complexity of a relational database increases, the skill set required by the RDBMS administrator, various users and report developers also increases. A mission-critical database may require expertise that exceeds the budget of a small business; furthermore, if the developers did not uniformly engage in best practice design, a subsequent developer may not understand hidden intricacies that could lead to broken queries or inaccurate reports. This risk increases if database and application development is performed by different people.

Hardware Performance

Complex queries require sophisticated processing power. Although most desktop computers can manage the databases of the size and complexity often encountered in a small business setting, a database with external data sources or very complex data structures may require more powerful servers to return results within an acceptable response time.